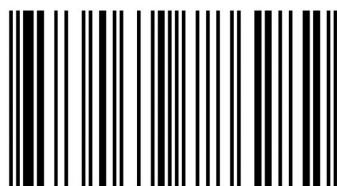


# Criptografia de curva elíptica - MAM em cloud

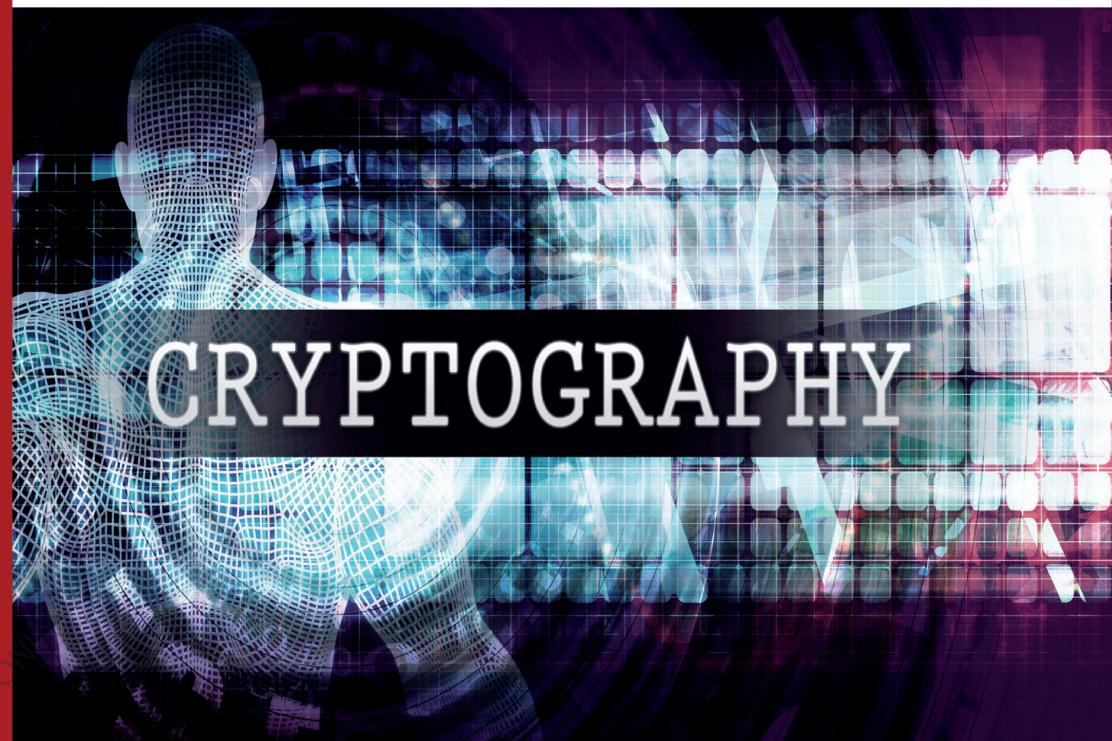
Neste livro é possível se ter uma visão sobre criptografia e sua utilização em um ambiente de broadcast de uma forma didática mesmo que o leitor não tenha familiaridade profunda sobre o tema. A criptografia é apresentada de forma didática, e a sua aplicação usando criptografia de curva elíptica em um ambiente onde a segurança e a resiliência são os pontos em destaque. Um sistema de MAM (Media Asset Manager), implementado em um ambiente em cloud é também descrito em detalhes tornando a leitura simples e estimulante para diversos públicos.

Mais de 20 anos em ambiente de Televisão – Broadcast TV Manchete, Rede TV, Rede Boas Novas, Bandeirantes, TV Educativa, TV Brasil. Gerente de Tecnologia – Grupo Frances Lesaffre. Gerente de Engenharia TV Escola e TV INES, a primeira emissora no Brasil voltada para o público surdo.

FOR AUTHOR USE ONLY



978-613-9-78628-2



Jorge Varella

## Criptografia de curva elíptica - MAM em cloud

Foco na segurança

Jorge Varella

**Criptografia de curva elíptica - MAM em cloud**

FOR AUTHOR USE ONLY



**Jorge Varella**

# **Criptografia de curva elíptica - MAM em cloud**

**Foco na segurança**

FOR AUTHOR USE ONLY

**Novas Edições Acadêmicas**

### **Imprint**

Any brand names and product names mentioned in this book are subject to trademark, brand or patent protection and are trademarks or registered trademarks of their respective holders. The use of brand names, product names, common names, trade names, product descriptions etc. even without a particular marking in this work is in no way to be construed to mean that such names may be regarded as unrestricted in respect of trademark and brand protection legislation and could thus be used by anyone.

Cover image: [www.ingimage.com](http://www.ingimage.com)

Publisher:

Novas Edições Acadêmicas

is a trademark of

International Book Market Service Ltd., member of OmniScriptum Publishing

Group

17 Meldrum Street, Beau Bassin 71504, Mauritius

Printed at: see last page

**ISBN: 978-613-9-78628-2**

Copyright © Jorge Varella

Copyright © 2019 International Book Market Service Ltd., member of  
OmniScriptum Publishing Group

FOR AUTHOR USE ONLY

*Esta dissertação é dedicada aos amigos que me apoaram  
e que ao final, ainda continuam meus amigos.*

# Agradecimentos

Agradecimentos principais são direcionados aos Professores: Prof. Dr. Renato Portugal, Prof. Dr. Bruno Richard Schulze, Prof. Dr. Giovane Quadrelli, Prof. Dr. Antonio Roberto Mury, Prof. Dr. Fabio Lopes Licht, Prof. Dr. Nelio Domingues Pizzolato, Prof. Dr. Pablo Javier Blanco, Prof. Dr. Robson Luiz Gaiofatto e todos aqueles que contribuíram para a produção deste trabalho acadêmico.

Agradecimentos especiais são também direcionados ao Centro de Engenharia e Computação, Programa de Pós-Graduação em Engenharia.

MESTRADO PROFISSIONAL EM GESTÃO DE SISTEMAS DE ENGENHARIA.<sup>1</sup> da Universidade Católica de Petrópolis.

FOR AUTHOR USE ONLY

---

<sup>1</sup> <<http://www.ucp.br/index.php/2015-07-13-20-14-42/pos-graduacao/stricto/mestrado-eng>>

FOR AUTHOR USE ONLY

*“Antes de fazer alguma coisa,  
pense, quando achar que já pode fazê-la ,  
pense novamente .  
(Pitágoras)*

# Resumo

O ambiente de *broadcast* tem um elevado custo na produção de programas a serem exibidos em sua grade de programação, é comum por conta das concessões de canais digitais terrestres, pequenas emissoras comunitárias deixarem de existir por não poderem manter conteúdo relevante e atrair audiência para seu canal. Esta dissertação visa demonstrar a implementação de um sistema em nuvem colaborativo de conteúdo áudio visual, de baixo custo e com segurança tanto de armazenamento de conteúdo local como em trânsito ou transporte, e que para isso faça uso de criptografia, e assim sendo que possa ser usado por emissoras comunitárias e públicas com poucos recursos disponíveis para produção de conteúdo. Por conta dos baixos recursos disponíveis o sistema proposto faz uso de softwares livres, como o sistema de gerenciamento de conteúdo (MAM) e o sistema de transcodificação. O sistema proposto em nuvem tem uma latência baixa por se esta uma premissa do transporte de conteúdo para os exibidores. A aplicação faz uso de um banco de dados em sua versão comercial por conta da criptografia de dados transparente e a necessidade de auditoria do banco de dados, estas características visam garantir um nível de segurança aceitável para os envolvidos no compartilhamento de conteúdo áudio visual.

**Palavras-chave:** TDE. Elíptica. Nuvem. FFMpeg. *OpenSource*.

# Abstract

The broadcast environment has a high cost in the production of programs to be displayed in its programming grid, it is common because of concessions of terrestrial digital channels, small community broadcasters cease to exist because they can not maintain relevant content and attract audience for your channel. This dissertation aims at demonstrating the implementation of a collaborative cloud system for low-cost and secure audio and visual content, both for local content storage and in transit or transport, using cryptography to make use of it. by community and public broadcasters with few resources available for content production. Due to the low resources available, the proposed system makes use of free textit softwares, such as the content management system (MAM) and the transcoding system. The cloud environment has a low latency if this is a premise of content transport. The application makes use of a database in its commercial version due to transparent data encryption and the need to audit the database, these features aim to guarantee an acceptable level of security for those involved in the sharing of visual audio content.

**Keywords:** TDE. Elíptica. Cloud. FFMpeg. OpenSource.

# **Lista de ilustrações**

Figura 1 – Virtualização-cloud . . . . .	20
Figura 2 – Clientes e serviços. . . . .	23
Figura 3 – Modelos cloud . . . . .	24
Figura 4 – Cloud Computing. . . . .	24
Figura 5 – Plataformas. . . . .	25
Figura 6 – Criptografia Simétrica. . . . .	30
Figura 7 – Criptografia Assimétrica. . . . .	33
Figura 8 – Criptografia RSA. . . . .	33
Figura 9 – Exemplo de Curva Elíptica. . . . .	41
Figura 10 – Curva Elíptica. . . . .	47
Figura 11 – Aplicação. . . . .	52
Figura 12 – Primeira Autenticação. . . . .	53
Figura 13 – Segunda Autenticação. . . . .	54
Figura 14 – Código Dupla Autenticação. . . . .	54
Figura 15 – Tela Amazon. . . . .	56
Figura 16 – Acesso ao sistema. . . . .	57
Figura 17 – Razuna Cadastro. . . . .	58
Figura 18 – Razuna Upload. . . . .	60
Figura 19 – Razuna Share. . . . .	61
Figura 20 – Razuna Share Pastas. . . . .	62
Figura 21 – Metadados. . . . .	63
Figura 22 – Razuna Codecs. . . . .	64
Figura 23 – Razuna Preview. . . . .	65
Figura 24 – Razuna S3 AWS. . . . .	67
Figura 25 – Razuna Backup. . . . .	68
Figura 26 – S3 AWS. . . . .	69
Figura 27 – TrasnCoder FFTrans. . . . .	70
Figura 28 – Pastas Mapeadas Windows. . . . .	71
Figura 29 – Playout. . . . .	72

Figura 30 – MySql TDE.	73
Figura 31 – Auditoria.	74
Figura 32 – ECC Autenticação AWS.	75
Figura 33 – ECC Razuna.	76
Figura 34 – ECC OpenSSL.	77
Figura 35 – ECC OpenSSL Certificado.	78
Figura 36 – Clientes Razuna.	80

FOR AUTHOR USE ONLY

# Lista de tabelas

Tabela 1 – Comparação de modelos . . . . .	18
Tabela 2 – Matriz de conceitos cruzados. . . . .	26
Tabela 3 – ECC ElGamal Criptografia. . . . .	43
Tabela 4 – ECC ElGamal Decriptografia. . . . .	44
Tabela 5 – ElGamal Decriptografia. . . . .	45
Tabela 6 – Comparação de tamanho de chaves <i>AES</i> , <i>ECC</i> e <i>RSA</i> . . . . .	46
Tabela 7 – Eficiência ECC. . . . .	48
Tabela 8 – RSA e ECC . . . . .	48
Tabela 9 – Nível de segurança. . . . .	49

FOR AUTHOR USE ONLY

# **Lista de abreviaturas e siglas**

AES	Algoritmo de criptografia de dados simétrico
RSA	Algoritmo de criptografia de dados Assimétrico.
ECC	Criptografía de Curvas Elípticas
ECDSA	Elliptic Curve Digital Signature Algorithm
ECDH	Elliptic curve Diffie–Hellman
ECMQV	Elliptic Curve Menezes-Qu-Vanstone
ECDLP	Elliptic Curve Discrete Logarithm Problem
NIST	National Institute of Standards and Technology
VT	Vídeo Tape Betacam
CODEC	Acrônimo de Codificador/Decodificador
LAG	Link Aggregation Group
SAAS	Software as a Service
IAAS	Infrastructure as a Service
PAAS	Plaform as a Service
NSA	National Security Agency
MAM	Media Asset Management
TDE	Transparent Data Encryption
ID	Identify
FFTrans	Transcoder para Broadcast

TLS      Transport Layer Security

SSL      Secure Sockets Layer

FOR AUTHOR USE ONLY

# Lista de símbolos

$\varphi$	Letra grega Phi
$\Gamma$	Letra grega Gama
$\Lambda$	Lambda
$\zeta$	Letra grega minúscula zeta
$\in$	Pertence

FOR AUTHOR USE ONLY

# Sumário

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>17</b>
1.1	Motivação . . . . .	18
1.2	Apresentação . . . . .	19
1.3	Objetivos . . . . .	19
1.4	Estrutura do Trabalho . . . . .	21
<b>2</b>	<b>AMBIENTE VIRTUAL . . . . .</b>	<b>22</b>
2.1	Virtualização . . . . .	22
2.2	Ambiente em nuvem . . . . .	22
2.3	Serviços em nuvem . . . . .	25
2.4	Nuvens Pública, Privada e Híbrida . . . . .	26
2.5	Segurança e auditoria . . . . .	27
<b>3</b>	<b>CRİPTOGRAFIA . . . . .</b>	<b>29</b>
3.1	<b>Algoritmos de Criptografia . . . . .</b>	<b>30</b>
3.1.1	Criptografia Simétrica . . . . .	30
3.1.2	Criptografia Assimétrica . . . . .	32
3.2	<b>Criptografia RSA . . . . .</b>	<b>33</b>
3.3	<b>Criptografia de Curva Elíptica . . . . .</b>	<b>39</b>
3.3.1	Características das curvas elípticas . . . . .	39
3.4	<b>Protocolo de criptografia em curva elíptica . . . . .</b>	<b>41</b>
3.4.1	Protocolo ECDSA . . . . .	42
3.4.2	Protocolo ECDH . . . . .	42
3.4.3	Protocolo ECMQV . . . . .	42
3.4.4	Protocolo ECDLP . . . . .	42
3.4.5	ElGamal . . . . .	42
3.5	<b>Tamanho de chaves RSA e curva Elíptica . . . . .</b>	<b>46</b>
3.6	<b>Protocolo ECDH - Diffie Hellman . . . . .</b>	<b>46</b>
3.7	<b>Comparação entre RSA e ECC. . . . .</b>	<b>48</b>

3.8	<b>Escolha da Curva Elíptica.</b>	49
4	<b>APLICAÇÃO</b>	51
4.1	Ambiente em Nuvem	53
4.2	MAM - Media Asset Management	57
4.3	Armazenamento	65
4.4	Transcoder FFTrans	69
4.5	Playout	71
4.6	Banco de dados	73
4.7	Criptografia Aplicada	75
5	<b>CONCLUSÃO</b>	79
5.1	Trabalhos Futuros	81
	<b>REFERÊNCIAS</b>	82
	<b>APÊNDICE A – PROGRAMA EM MAPLE - CRIPTOGRAFIA RSA.</b>	87
	<b>APÊNDICE B – PROGRAMA EM MAPLE - IMPLEMENTAÇÃO ECC - ELGAMAL4.MPL</b>	89
	<b>APÊNDICE C – PROGRAMA EM MAPLE - MÉTODO DE CRIPTOGRAFIA ECC</b>	92
	<b>APÊNDICE D – PROGRAMA EM MAPLE - CURVAS ELÍPTICAS.</b>	96

# 1 INTRODUÇÃO

Durante muitos anos o ambiente de *broadcast* teve uma pequena participação de computadores nas suas operações. No inicio dos anos de 1990, algumas emissoras de TV pelo mundo começaram a fazer uso de sistemas informatizados para controlar dispositivos, tais como *Videotapes*, (VTs). A *Bloomberg*, emissora de Tv norte americana foi uma das pioneiras no uso de computadores, que além de controlar os dispositivos, passou também usa-los na operação de controle e exibição de conteúdo. Neste mesmo período os primeiros softwares para edição de áudio e vídeo surgem. Os sistemas informatizados começam a fazer parte efetivamente do ambiente de *broadcast*, nos anos seguintes, sistemas de armazenamento e exibição foram criados para melhorar o fluxo de trabalho centralizado que existiam e que ainda são usados por algumas das emissoras de TV.

Com a digitalização de conteúdos audiovisuais, novos processos de exibição foram criados como, por exemplo exibidores digitais de conteúdos. A digitalização do legado analógico se torna uma condição indispensável para prover conteúdo aos exibidores, pois com o formato original do material, criou-se uma grande dificuldade de exibição. Novos *codecs* de video surgiram para atender as diversas plataformas, assim como um público que passou a assistir a TV, de forma diferente da tradicional, ou seja a forma linear de exibição, este público quer assistir seus programas favoritos em horários diversos e com isso novas tecnologias foram usadas neste sentido. A virtualização foi uma tendência natural de migração de diversos sistemas, onde a rapidez em construir os ambientes para atender a uma demanda temporária ou a um nível de processamento muito elevado por um período de tempo curto. Ambientes virtualizados de transcodificação de video, sistemas de inserção de caracteres, logos animados entre outros, impulsionaram a utilização deste tipo de ambiente, com um custo menor e mais ágil na sua implementação.

O passo seguinte foi a transferência deste ambiente virtual para o ambiente em nuvem, e que se mostrou novamente uma tendência de evolução natural para vários ambientes informatizados dentro do setor televisivo, porém no ambiente de *broadcast* existem muitos detalhes técnicos, tais como latência, segurança, sistemas de backup próprio, etc, os quais serão abordados ao longo desta dissertação.

Na tabela 1 é possível comparar os ambientes e a evolução dos mesmos em um cenário onde a evolução tecnologia se faz presente.

Tabela 1 – Comparaçāo de modelos

Modelo analógico	Modelo Digital	Modelo Digital
Redundância Limitada	Redundância Simplificada	Redundância Simplificada
Pouco uso da Informática	Uso da Informática	Uso da Informática
Custo elevado na atualização	Custo reduzido comparado ao modelo analógico	Custo reduzido comparado a modelo Digital
Centralizado	Centralizado	Descentralizado

Fonte: Elaborada pelo autor.

## 1.1 Motivação

Manter o parque de equipamentos atualizados e alinhados com as tecnologias mais modernas é um desafio para qualquer negócio, porém quando isto envolve um custo elevado esta dificuldade aumenta em muito, no ambiente de *broadcast* substituir um servidor de vídeo ou apenas a substituição de um *codec*, pode envolver um custo de milhares de dólares. Uma das primeiras soluções eficientes para se manter alinhado com o mercado televisivo digital, foi a virtualização, este novo paradigma tornou o processo de atualização de *hardware* e *software*, muito mais eficiente e menos oneroso. Para entender como isso acontece faz-se necessário entender os processos básicos de produção, edição e exibição de conteúdo audiovisual em um ambiente televisivo.

Será tomado como exemplo uma emissora *tapeless*. Ou seja um ambiente onde não há tráfico de mídia física, a captação do material audiovisual é feita por câmeras em banda base, ( áudio e vídeo ) estes sinais são direcionados a um sistema de conversão e armazenamento que é chamado de *ingest*, este processo de conversão gera uma arquivo com um *codec* específico e suas configurações, e que será armazenado nos servidores de vídeos. Este material bruto deve ser disponibilizado para o ambiente de edição através de redes que se possa trafegar arquivos de video de alta resolução, as redes usando fibra se apresentam com a melhor solução, embora redes *ethernet* em *lag* também possam ser usadas. Com o material editado este retorna para os servidores de exibição, onde serão catalogados e posteriormente exibidos pelos sistemas denominados *playout* ou exibidores.

Estes processos interdependentes torna a operação rápida e eficiente, porém qualquer alteração neste fluxo pré configurado acarretará impacto no processo dependente. Neste momento a virtualização torna-se uma ferramenta imprescindível para realização de qualquer mudanças. Provisionar uma máquina e deixá-la operacional para a tarefa é algo de baixa complexidade em um ambiente virtual, porém determinadas tarefas são executadas em curto espaço de tempo e envolvem grande poder

computacional, isto demanda um grande investimento em hardware e que passa uma grande parte do tempo ocioso, além disso, quando se tem um grupo de colaboradores externos que fazem uso destes serviços em tempos não predefinidos muitas vez coincidentes, e que demandam grande poder computacional, armazenamento, nível de segurança elevado, a questão passa ser: Como atender a esta demanda com eficiência, elasticidade, baixo custo e com segurança capaz de ser aceita pelos parceiros envolvidos?

## 1.2 Apresentação

Com o intuito de torna possível uma maior capilaridade na captação e distribuição de conteúdo audiovisual, a computação em nuvem ou *cloud computing*, se apresenta como a solução eficiente, de menor custo, comparado aos modelos tradicionais e ainda capaz de tornar possível, que pequenas emissoras, (TV comunitárias) sejam capazes de produzir e distribuir conteúdo para diversas praças sem necessariamente ter que desembolsar grandes quantias na compra de equipamento e ainda ter pessoal treinado capaz de operacionalizar as diversas funções que podem ser no ambiente em nuvem automatizadas. Um exemplo pode ser visto na figura 1, onde a produção de conteúdo é enviada para nuvem e exibido onde for permitido. Além da contribuição para programação ser feita de diversos pontos.

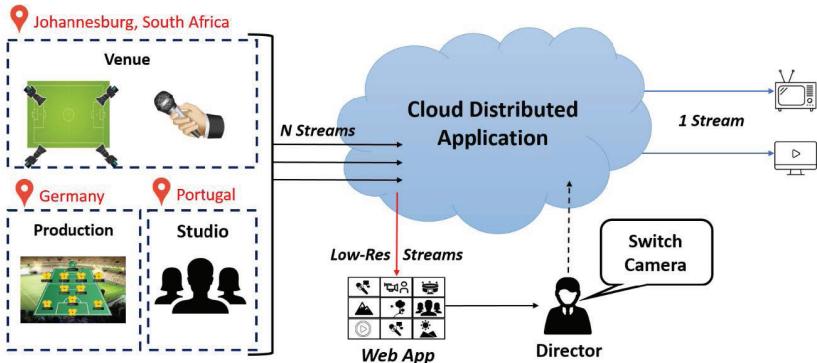
Na figura 1 ainda pode-se ver o material sendo gerado por diversas fontes e sendo enviados para o ambiente em nuvem, uma conversão para um único formato de vídeo deve ser realizado para evitar necessidade de novas transcodificações fora deste ambiente. Contudo vale ressaltar que uma catalogação mínima é exigida onde os metadados devem ser accordados entre todos os envolvidos.

## 1.3 Objetivos

Este trabalho de dissertação tem como objetivo principal propor um modelo de *playout* usando o ambiente em nuvem para a área de *broadcast*. Este modelo serve de base teórica para futuras aplicações em escalas maiores, além de ter a finalidade de servir como ponto de referência para futuras aplicações práticas.

Em alinhamento com o objetivo principal, o ambiente deve dispor de recursos capazes de satisfazer determinadas condições, como um serviço de transcodificação de vídeo para um formato acordado entre as partes envolvidas, sem que para isso aja intervenção manual ou seja, basta o

Figura 1 – Virtualização-cloud



Fonte: Cunha (2016).

colaborador depositar o material audiovisual em uma pasta, para que o conteúdo seja convertido em um formato que poderá ser usados por todos os participantes deste compartilhamento.

Uma base de dados contendo metadados é necessária para a catalogação do material, esta base será usada por todos os colaboradores na situação de busca, exibição e *download* de material, o arquivo original somente estará disponível para o *owner*.

A segurança do ambiente deverá garantir que o acesso somente poderá ser feito por pessoas autorizadas e todo o conteúdo deverá estar criptografado usando o algoritmo de criptografia de curva elíptica, a fim de manter o mesmo protegido contra cópias não autorizadas e acessos indevidos que por ventura venham a ocorrer.

Assim este trabalho visa contribuir para a disseminação da tecnologia em um ambiente em nuvem, em um seguimento corporativo que está dando os primeiros passos nesta direção, além de poder tornar possível a utilização compartilhada de recursos computacionais entre pequenas emissoras de TV que sozinhas não teriam condições de fazê-lo.

## 1.4 Estrutura do Trabalho

No capítulo 2 é apresentado o ambiente em nuvem, seus vários tipos de serviços, definição de nuvem pública e privado cujas características serão explicitadas, também será mostrado a aderência do ambiente em nuvem ao tipo de negócio. Ao final do capítulo a segurança e auditoria serão abordados.

No capítulo 3 a criptografia será o foco. A criptografia de chave pública *RSA* e seus vários protocolos como *ECDSA*, *ECDH*, serão descritos, além de um exemplo prático usando o algoritmo . A criptografia de curva elíptica, suas características matemáticas também são demostradas neste capítulo. Uma comparação entre criptografia de curva elíptica e criptografia RSA, no que tange a tamanho de chave e a escolha da curva finalizam este capítulo.

O capítulo 4 é dedicado a escolha dos itens que compõem o projeto e suas características técnicas e operacionais. Estes itens foram divididos em:

- Ambiente em nuvem.
- MAM.
- Armazenamento em nuvem.
- Transcodificação
- *Playout*.
- Banco de dados.
- Criptografia.

Por fim o capítulo 5 apresenta as conclusões obtidas no desenvolvimento desta dissertação.

## 2 AMBIENTE VIRTUAL

### 2.1 Virtualização

A utilização da virtualização de recursos computacionais nos permite uma série de benefícios segundo Yokoyama (2015), dentre eles é possível destacar: Melhor utilização dos equipamentos existentes de forma a permitir maior flexibilidade, visto que um mesmo *hardware*, pode ser utilizado por várias aplicações e pode inclusive executar sistemas operacionais diferentes; a segurança do ambiente é significativamente aumentada por conta da separação de problemas nas máquinas virtuais, assim sendo um problema em uma máquina virtual não se propaga a outras máquinas virtuais no mesmo *host*.

De acordo com Xing e Zhan (2012), a nuvem depende intrinsecamente da virtualização, a grande maioria dos benefícios mencionados anteriormente somente tornam-se possíveis por conta da virtualização. Avanços relevantes nesta área e a redução de custo da camada de virtualização despertaram o interesse de diversos setores, incluindo o setor televisivo.

Uma máquina virtual segundo Popek e Goldberg (1974), pode ser definida como uma duplicata eficiente de uma máquina real, e que pode ser controlada por *software* ou *hardware*, embora a virtualização não seja o foco principal desta dissertação é preciso que seus conceitos sejam apresentados.

### 2.2 Ambiente em nuvem

O modelo de nuvem tem sua origem no uso de grandes centros de dados, capazes de tratar serviços Web em grande escala na Internet. A sua ideia não é nova, porém sua adoção possibilitou o uso de forma eficiente dos recursos de software e hardware. O gerenciamento automatizado, técnicas de balanceamento de carga e virtualização tornaram possível a alocação dinâmica de recursos e a elasticidade de provisionamento da infraestrutura contratada pelo cliente. Pode-se dividir a arquitetura de nuvem em quatro camadas de acordo com Zhang, Cheng e Boutaba (2010): Centro de dados, infraestrutura, plataforma e aplicação. Todas podem ser vistas como serviço para a camada acima; e como cliente para camada abaixo. Esta arquitetura alcançou popularidade pela oferta da infraestrutura de nuvem em três principais modelos de serviço.

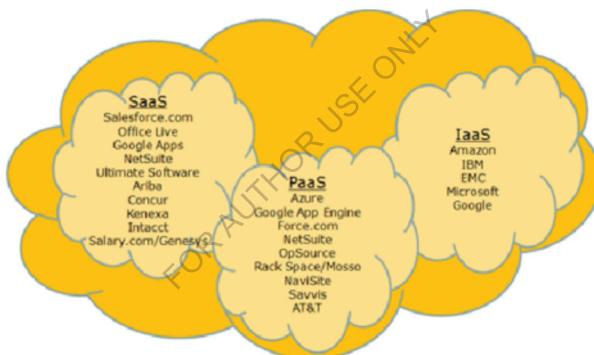
- Software como Serviço – *Software as a Service, SaaS*.
- Plataforma como Serviço – *Platform as a Service – PaaS*.
- Infraestrutura como Serviço – *Infrastructure as a Service – IaaS*.

Na figura 2 é possível ver os vários tipos de clientes e serviços.

No ambiente em nuvem é possível encontrar diversos modelos, tais como:

- Nuvem Privada – *Private Cloud*.
- Nuvem Comunitária – *Community Cloud*.
- Nuvem Pública – *Public Cloud*.
- Nuvem Híbrida – *Hybrid Cloud*.

Figura 2 – Clientes e serviços.

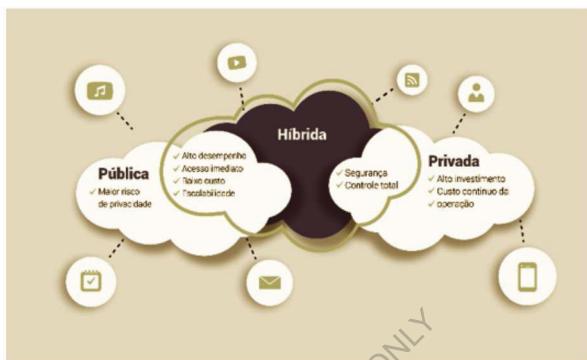


Fonte: Hwang, Dongarra e Fox (2013).

Existem vários modelos com suas vantagens e desvantagens, porém não serão o foco desta dissertação. A opção por determinado modelo dependerá do cenário específico, considerando o modelo de negócio do cliente, estes modelos de nuvem podem ser vistos na figura 3. A popularidade e o nível de amadurecimento podem ser medidos pelo número de serviços de nuvem ofertados por diversas empresas, que se especializaram neste tipo de solução tecnológica. Em alguns casos, não era o *core* de

seu negócio, como a *Amazon* por exemplo, em que o ambiente foi criado para possibilitar a sua rápida expansão em determinados meses, quando a demanda era muito maior. Nos meses com demanda menor, passou-se a ofertar ao mercado a capacidade ociosa de seus sistemas.

Figura 3 – Modelos cloud



Fonte: Hwang, Dongarra e Fox (2013).

Um modelo básico de nuvem pode ser visto na figura 4. Este modelo busca evidenciar as várias formas de colaboração de diversos dispositivos, serviços e modelos.

Figura 4 – Cloud Computing.



Fonte: Hwang, Dongarra e Fox (2013).

## 2.3 Serviços em nuvem

Segundo Amazon (2017), existem três modelos principais de computação em nuvem e cada modelo representa uma parte da computação em nuvem.

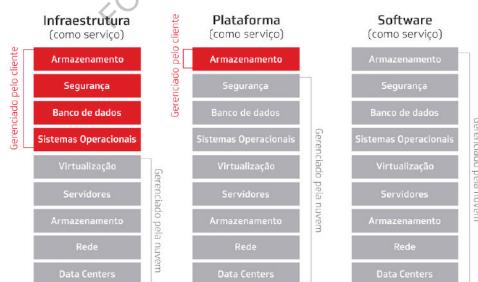
O modelo *IaaS*, possui os componentes básicos da tecnologia da informação em nuvem, predominantemente da acesso aos recurso de rede e computadores, além do espaço de armazenamento. A estrutura como um serviço, oferece um alto nível de flexibilidade e controle de gerenciamento sobre os recursos de TI, com os quais suportes e desenvolvedores estão familiarizados.

O modelo *SaaS*, oferece aplicativos via *web*. Este serviço é oferecido aos usuários sob demanda e funcionam através de um modelo de licenciamento, o provedor hospeda o aplicativo em sua infraestrutura. Um serviço *SaaS* deve oferecer um aplicativo ou *suite* sob demanda, atendendo usuários individuais ou várias organizações.

De acordo com Krutz e Vines (2010), o modelo *PaaS*, é similar ao modelo *SaaS*, porém é um ambiente completo para desenvolvimento de aplicativos e não apenas para sua utilização. O principal diferencial esta no tocante a fornecer uma plataforma virtual completa acessível via navegador *web*, com isso é possível acelerar consideravelmente o tempo de desenvolvimento e implantação de aplicativos.

Pode-se ver as características na figura 5.

Figura 5 – Plataformas.



Fonte: Adaptado de Hwang, Dongarra e Fox (2013).

Com suas características citadas é possível construir uma matriz de conceitos cruzados dos três modelos que pode ser visto na tabela 2.

Tabela 2 – Matriz de conceitos cruzados.

	Mudança de paradigma	Caracte-rísticas	Palavras-Chaves	Vantagens	Riscos	Quando não usar
IaaS	Infra-estrutura como um ativo	Geralmente independente da plataforma; custos de infraestrutura são compartilhados ; SLAs; pagamento pelo uso; escalamento automático	Computação em grade, instância de computação, <i>hypervisor</i> , <i>cloudbursting</i> , agrupamento de recursos	Evita despesa de capital com hardware e RH; risco de ROI reduzido; barreiras pequenas à entrada; escala simples e automatizada	Eficiência e produtividade ; custo de longo prazo potencialmente maior; Requer medidas de segurança novas ou diferentes	Quando orçamento capital é maior que o orçamento operacional
PaaS	Compra de licença	Consome infraestrutura da nuvem; voltado a métodos de gerenciamento de projeto ágeis	Pilha de solução	Implementação de versão simplificada	Centra-lização requer medidas de segurança novas ou diferentes	Não disponível
SaaS	Software como um ativo	SLAs; UI construída com aplicativos de thin client; componentes de nuvem; comunicação via APIs; conectadas vagamente; modular	<i>Thin client</i> ; aplicativo cliente ou servidor	Evitar gasto de capital ; risco de ROI reduzido; atualizações simplificadas	Requer medidas de segurança novas ou diferentes	Não disponível

Fonte: IBM (2017).

## 2.4 Nuvens Pública, Privada e Híbrida

Nesta dissertação destaca-se-á apenas dois modelos de infraestrutura a pública e a privada, pois sua utilização terá grande relevância na sua aplicação prática. A infraestrutura em nuvem pública pode ser definida segundo Krutz e Vines (2010) como sendo disponibilizada para o público em geral e infraestrutura de nuvem privada aquela que é operada apenas por uma organização. No que tange a segurança é possível afirmar que o ambiente em nuvem privada é mais seguro que o ambiente em

nuvem pública, sendo que neste ambiente privado a infraestrutura é operada pela própria organização que a criou.

Pode-se implementar qualquer um dos modelos, verificando qual deles oferece a melhor solução, ainda segundo Krutz e Vines (2010), um projeto temporário, por exemplo, pode ser melhor em uma nuvem pública, por uma relação direta do custo da aquisição de equipamentos para um demanda temporária.

Vale ressaltar que os modelos não especificam localização da infraestrutura ou aplicação e podem ser co-localizadas.

## 2.5 Segurança e auditoria

A computação em nuvem pode ser vista segundo Wang et al. (2010), como uma arquitetura de tecnologia da informação e devido as suas características tais como, acesso onipresente, utilização sobre demanda, rápida elasticidade de recursos entre outros, vem transformando como as empresas usam a tecnologia da informação. Um dos principais aspectos dessa mudança esta na condição em que os dados são armazenados por terceiros, está forma de armazenamento gera um série de benefícios, tais como alívio da carga para gerenciamento de armazenamento, acesso universal aos dados com localização geográfica independente, além de evitar investimentos em software, hardware e equipe de suporte. Embora a computação em nuvem possua uma série de vantagens, a mesma possui também uma grande quantidade de ameaças a segurança aos dados de usuários em um ambiente terceirizado.

De acordo com Juels e Jr (2007), como os usuários não possuem os dados fisicamente armazenados, as primitivas criptográficas não podem ser adotadas diretamente, a simples verificação de integridade de uma arquivo pode ser uma tarefa complexa por conta de custo de transferência em toda a rede, o processo de criptografia em ambiente nuvem, será destacado no próximo capítulo, assim como suas características.

Segundo Wang et al. (2009), considerando a grande dimensão dos dados terceirizados e a capacidade limitada de recurso do usuário final, as tarefas de auditoria de dados em um ambiente em nuvem podem ser dispendiosas e extremamente lentas para os usuários.

Em um contexto de garantir a integridade de dados armazenados remotamente por terceiros, recentemente tem se proposto a noção de auditabilidade pública, esta por sua vez permite que um auditor externo, além do próprio usuário verifique os dados. Porém segundo Wang et al. (2010), a

proteção de privacidade para com os auditores externos e relegado a segundo plano, embora seja uma falha na segurança a possibilidade de se revelar informações de dados dos usuários por terceiros. Explorar o uso da criptografia antes da transferência do conteúdo é uma forma de mitigar a preocupação com a privacidade dos dados e assim como mencionado anteriormente será tema do próximo capítulo.

FOR AUTHOR USE ONLY

## 3 CRIPTOGRAFIA

Manter em sigilo informações é algo que o ser humano se preocupa a milhares de anos. Manter as informações seguras se torna cada dia mais vital, disso depende a governabilidade de países e a continuação dos negócios das empresas. Imagine que por falta de segurança em uma empresa seus segredos fossem roubados por concorrentes, isto seria desastroso para qualquer corporação. O usuário comum também compartilha dessas preocupações quando os seus dados pessoais podem se tornar expostos.

A segurança da informação sofreu mudança com a evolução das tecnologias, antes as informações eram transcritas para meios físicos e guardados em armários robustos ou cofres para que não fossem roubados. Com o advento dos computadores essas informações passaram a ser transformadas em dados digitais, este novo formato fez com que novas técnicas fossem criadas com o intuito de preservar a sua confidencialidade.

As redes locais e a internet trouxeram à tona um novo problema com o compartilhamento de informações entre os computadores. A segurança da Inter-rede (termo pouco empregado), se torna fundamental para o mundo virtualizado, visto que praticamente todas as empresas, organizações acadêmicas e até mesmo os usuários domésticos usam equipamentos interconectados nas transações de processamentos de dados. Com a motivação da segurança das informações foram criadas cifras, técnica para mascarar uma mensagem de modo que somente os destinatários habilitados conseguissem compreender corretamente seu conteúdo. Segundo Alecrim (2005), o termo criptografia surgiu da fusão das palavras gregas "kryptós" e "gráphein", que significam "oculto" e "escrever", respectivamente. A origem da criptografia, segundo dispõem Loidreau e Sendrier (2001), provavelmente, remonta aos princípios da existência humana, logo que as pessoas tenham tentado aprender a se comunicar e por estar ligada de modo inerente ao oculto na sua vertente mais concreta. Consequentemente, segundo o autor em comento, os seres humanos tiveram de encontrar meios para garantir a confidencialidade de parte das suas comunicações, e a origem desses meios ou estratégias se encontra na criptografia.

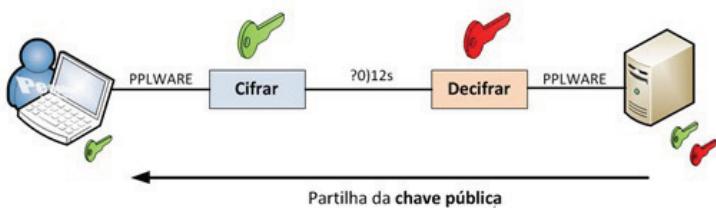
Com a evolução da tecnologia, Silva, Macharet e Teixeira (2008) expõem e disserta acerca da existência de diversos algoritmos e protocolos de criptografia complexos que promovem a proteção e a segurança de uso e exposição de dados de forma eficiente.

## 3.1 Algoritmos de Criptografia

### 3.1.1 Criptografia Simétrica

É o tipo de criptografia em que o emissor e o receptor usam a mesma chave para criptografar e deccriptografar a mensagem (chave compartilhada), como mostra a Figura 6.

Figura 6 – Criptografia Simétrica.



Fonte: [dusted.codes](http://dusted.codes) (2015).

A cifra é definida sobre um par de algoritmos de encriptografia ( $E$ ) e desencriptografia ( $D$ ), onde:

$$c = E(k, m) \text{ e } m = D(k, c)$$

$m$  = Texto claro

$c$  = Texto cifrado

$k$  = chave criptografada

$E$  = algoritmo de encriptação

$D$  = Algoritmo de desencriptação

$m = D(k, E(k, m))$ , Usando a Equação de consistência:

$$m \in M, k \in K : D(k, E(k, m)) = m$$

Para toda mensagem pertencente ao espaço de mensagem (informação) e toda chave pertencente ao espaço de chave o resultado da operação de consistência será a mensagem decriptada. Os modelos

de criptografia de chave simétrica são baseados em dois princípios, substituição e transposição.

Substituição: Cada elemento do texto claro é mapeado em outro elemento.

Transposição: Elementos do texto claro são reorganizados.

Na criptografia assimétrica destaca-se o modelo *AES* que teve sua criação em Janeiro de 1997 a pedido do **NIST** (*National Institute of Standards and Technology*), órgão do departamento de comércio dos Estados Unidos encarregado de aprovar padrões, e que decidiu que precisava de um novo padrão e patrocinou uma competição de criptografia. Pesquisadores do mundo todo participaram, este novo padrão seria chamado de *AES* (*Advanced Encryption Standard*), as regras para a competição foram as seguintes:

O Algoritmo em cifra de bloco simétrico.

Projeto público.

Chaves de 128, 192 e 256 bits.

Poderia ser implantado tanto em software com em hardware.

O algoritmo também deveria ser público.

Em outubro de 2000 foram sagrados vencedores os Belgas, Joan Daemen e Vicent Rijmen, com o projeto Rijndael, seu nome é derivado do sobrenome dos autores e tornou-se padrão em novembro de 2001. Como *DES*, o *AES* também utiliza substituição e permutação, além de empregar varias rodadas. O número de rodadas depende do tamanho da chave e do bloco, sendo 10 para cada chave de 128 bits com blocos de 128 bits, passando para 14 no caso de chave ou bloco maior. No entanto todas as operações envolvem bytes inteiros diferente do *DES* (Algoritmo de criptografia cujas a vulnerabilidade estava provada), e assim permitir implementações eficientes, tanto em software com em hardware.

Em MORENO PEREIRA (2005), o *AES* é classificado como um cifrador de bloco com tamanho de chave que pode variar entre os valores de 128, 192 e 256 bits, o que significa que se pode ter tamanho de blocos e chaves diferentes. Em função do tamanho de bloco e chaves, determina-se a quantidade de rodadas necessárias para cifrar e decifrar. O *AES* opera com um determinado número de 32 bits, que são ordenados em colunas de 4 bytes denominados *Nb*. Os valores possíveis são de 4, 6, e 8 equivalentes a blocos de 128, 192 e 256 bits. Por isso sempre que *Nb* for referido, significa que se tem  $Nb \times 32$  bits de tamanho de blocos de dados. A chave é agrupada da mesma forma que o bloco de dados, isto é, em colunas, sendo representado pela sigla *Nk*. Com base nos valores que *Nb* e *Nk* podem assumir é que se determina a quantidade de rodadas a serem executadas, identificada

pela sigla *Nr*. No processo de cifragem e decifragem usa-se as funções mais comuns nos cifradores. Cada bloco ou estado é sujeitado durante o processo para cifrar as seguintes iterações:

*SubByte*: os bytes de cada bloco são substituídos por seus equivalentes em uma tabela de substituição (S-BOX);

*ShiftRow*: ou deslocamento de linha: nesta etapa, os bytes são rotacionados em grupos de 4 bytes;

*MixColumn*: cada grupo de 4 bytes sujeita-se a uma multiplicação modular, o que proporciona a cada byte do grupo influenciar todos os outros bytes;

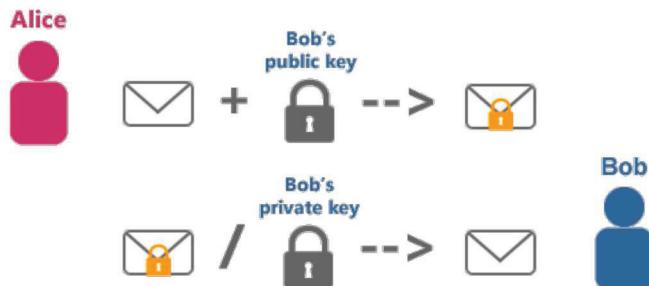
*AddRoundKey*: ou adição de chave de rodada: nesta fase, o bloco de dados é alterado por meio da sub-chave da rodada, a qual possui o mesmo tamanho do bloco, que realiza uma operação XOR com blocos inteiros.

### 3.1.2 Criptografia Assimétrica

A criptografia assimétrica usa duas chaves distintas, de modo a obter uma comunicação segura através de canais de comunicação inseguros. Também conhecido como algoritmo de chave pública, nasceu com o intuito de resolver o problema de distribuição de chaves. Segundo Tanenbaum (2003), em 1976 dois pesquisadores da *University of Stanford, Diffie e Hellman*, propuseram um algoritmo de criptografia completamente novo, no qual as chaves de criptografia e de descriptografia eram diferentes, e a chave de descriptografia não podia ser derivada da chave de criptografia. O Algoritmo tinha três requisitos básicos:  $D(E(P)) = P$ , aplicado  $D$  a uma mensagem criptografada,  $E(P)$ , seria obtido outra vez um texto simples  $P$ . Isso permite que o destinatário legítimo decodifique o texto cifrado. É extremamente difícil deduzir  $D$  a partir de  $E$ . ( $E$ ), não pode ser decifrado por um ataque de texto simples. No algoritmo usa-se uma chave pública para criptografar e uma chave privada para descriptografar. Este algoritmo de criptografia será detalhado no capítulo seguinte.

Uma visão mais didática sobre criptografia assimétrica pode ser vista na figura 7. Pode-se destacar o processo de criptografia e descriptografia da mensagem entre os interlocutores.

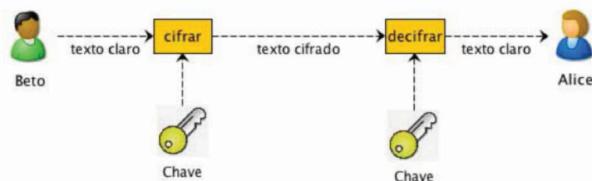
Figura 7 – Criptografia Assimétrica.

Fonte: [dusted.codes](http://dusted.codes) (2015).

## 3.2 Criptografia RSA

A criptografia e suas operações, métodos e estratégicas, segundo Garfinkel e Spafford (1999), representam um conjunto de técnicas que são usadas para que se mantenha a informação segura. Estas técnicas consistem na utilização de chaves e algoritmos de criptografia, um desses algoritmos é o **RSA**, que possui desenvolvimento em chaves públicas. O algoritmo RSA implementa um sistema assimétrico ou o uso de uma chave pública. Como pode ser visto na figura 8, tendo conhecimento da chave e do algoritmo usado é possível desembaralhar e ler a mensagem recebida.

Figura 8 – Criptografia RSA.

Fonte: [dusted.codes](http://dusted.codes) (2015).

No processo de codificação e decodificação, as chaves apresentadas na figura 8, são utilizadas para transformar o texto original em um texto criptografado e vice-versa, ou seja, também é possível a decifração do que foi criptografado, tudo isto por meio de chaves e no caso da criptografia RSA essas referidas chaves são públicas. A figura 8, oferece ainda uma definição didática-visual de como ocorre o processo de criptografia baseada em RSA, trata-se, portanto, de um esquema sintético de como se dá a criptografia, o texto original é cifrado pelo remetente da comunicação por uma chave pública, só será possível a leitura e entendimento da mensagem recebida se o destinatário tiver de posse da referida chave privada, que irá decodificar o conteúdo da mensagem. Este algoritmo, o RSA, foi desenvolvido pelos professores do MIT, Ronald Rivest e Adi Shamir e o professor da *USC Univesity of Southern Califórnia*, Leonard Adleman, é o mais usado e fácil de implementar dos algoritmos assimétricos seu nome é formado pelas iniciais dos seus desenvolvedores, Rivest, Shamir e Adleman, e tem como base a dificuldade de fatorar números inteiros. Este sistema de criptografia consiste em gerar uma chave pública usada para criptografar os dados e uma chave privada para descriptografar, as chaves devem ser números primos grandes, o que dificulta a obtenção de uma chave a partir de outra. Assim, a técnica de criptografia RSA utiliza o problema da fatoração de dois números primos grandes, como base de sua segurança. A segurança deste tipo de criptografia, como já mencionado, depende do tamanho do número primo fornecido, ou seja, quanto maior for esse número maior a segurança.

Normalmente utiliza-se números primos de 512 bits que combinados formam chaves de 1024 bits, em aplicações bancárias podem chegar a 2048 bits. Os algoritmos são ainda divididos em dois tipos básicos segundo menciona BARBOSA (2003): "Existem os algoritmos específicos e algoritmos genéricos". Algoritmos específicos, segundo o autor, são utilizados em situações e casos bem definidos e já delineados, baseando-se em determinados aspectos, ou seja, onde já se conhece um determinado tipo de parâmetro, mas ficando o seu uso restrito a esses tipos de situações bem estruturadas. Já os algoritmos genéricos não se preocupam com qualquer tipo de restrição, podem, desta forma, serem usados nas mais diversas situações criptográficas.

Os algoritmos para geração de chaves públicas e privadas são relativamente simples de utilização e entendimento, com base nas teorizações é possível desenvolver um sistema criptográfico empresarial ou doméstico, para uso eficiente, sem lacunas e gerando grande segurança nos processos informacionais.

De acordo com Rivest, Shamir e Adleman (1978), citado em MORENO PEREIRA (2005), mostra-se a simplicidade para a geração destas:

- 1 – Escolhem-se dois números primos grandes  $p$  e  $q$ .
- 2 – Gera-se um número  $n$  multiplicando-se  $p$  e  $q$ ,  $n = p \times q$ .
- 3 – Escolhe-se um número  $d$ , tal que  $d$  é menor que  $n$  e relativamente primo a  $(p - 1) \times (q - 1)$ .
- 4 – Escolhe-se um número  $e$ , tal que  $(ed - 1)$ , seja divisível por  $(p - 1) \times (q - 1)$ .
- 5 – Os valores  $e$  e  $d$  são chamados de expoentes públicos e privado respectivamente.

O par  $(n, e)$  é a chave pública e o par  $(n, d)$ , a chave privada.

Os valores  $p$  e  $q$  devem ser mantidos em segredo ou destruídos.

O algoritmo ou as chamadas chaves de RSA, segundo Rivest, Shamir e Adleman (1978), são de uso constante nos sistemas atuais de criptografia assimétrica. O processo delineado anteriormente, pode ser resumido, para melhor entendimento geral:

O interlocutor **A** escolhe secretamente, dois números primos,  $p$  e  $q$ , e pública  $n = pq$ . Então, **A** escolhe aleatoriamente  $b$ , tal que  $b$  e  $\varphi = (p - 1)(q - 1)$  são primos relativos ou primos entre si. **A** calcula  $a$  tal que  $ab = 1 \pmod{\varphi}$ . Sua chave secreta é  $a$ , enquanto que  $b$  é revelado publicamente. Interlocutor **B** cifra sua mensagem  $x$  computando:

$$y = x^b \pmod{n}.$$

e envia  $y$  para **A**, e então decifra e obtém  $x$  calculando:

$$x = y^a \pmod{n}.$$

No exemplo a seguir escolhe-se aleatoriamente dois números primos, para facilitar o acompanhamento do processo usar-se números pequenos. Estes números são identificados por  $p$  e  $q$ , onde  $(p = 19)$  e  $(q = 23)$ .

Agora calcular-se a chave pública com os números primos escolhidos:

$$\begin{aligned} n &= pq, \\ n &= 19 \times 23, \\ n &= 437. \end{aligned}$$

A seguir calcula-se o  $\varphi(n)$ :

$$\begin{aligned}\varphi(n) &= (p-1) \times (q-1), \\ \varphi(n) &= (19-1) \times (23-1), \\ \varphi(n) &= 18 \times 22 = 396.\end{aligned}$$

Agora para encontrar-se o  $e$ , é necessário encontrar um número que seja relativamente primo de  $\varphi(n)$ . Fatorando o  $\varphi(n) = 396$ , obtém-se o seguinte:

$$396 \mid 2$$

$$198 \mid 2$$

$$99 \mid 3$$

$$33 \mid 3$$

$$11 \mid 11$$

$$1 \mid$$

Ou seja,  $396 = 2 \times 2 \times 3 \times 3 \times 11$ .

Para que  $e$  e 396 sejam primos o valor de  $e$ , não pode ser divisível por 2, 3 e 11. Escolhe-se o número 13, mas para que se possa verificar se o número é valido sera necessário fazer o MDC de 13 e 396, e o resultado deve ser 1.

$$\begin{aligned}q_1 &= 396/13 = 30, \text{ resto } 6. \\ q_2 &= 13/6 = 2, \text{ resto } 1. \\ q_3 &= 6/2 = 3, \text{ resto } 0.\end{aligned}$$

Sendo assim, MMC (13, 396) é 1, ou seja, o último resto antes do 0. Assim sendo o par de chaves públicas é  $(e, n) = (13, 437)$ .

Para o cálculo do par de chaves privadas será usado o algoritmo de Euclides estendido:

- (1)  $13/396 = 0$ , resto 13 divide-se o valor pelo modulo.
- (2)  $396/13 = 30$ , com resto 6 ... divide-se o divisor anterior pelo resto.
- (3)  $13/6 = 2$ , com resto 1 ... divide-se o divisor anterior pelo resto.
- (4)  $6/2 = 3$ , com resto 0 ... divide-se o divisor anterior pelo resto.

O algoritmo de Euclides estendido usa apenas os restos diferentes de zero das divisões do cálculo do MMC (13 , 396).

$$13 = (1 \times 13)(0 \times 396).$$

$$13 = (1 \times 13).$$

$$6 = (1 \times 396) - (30 \times 13) \dots \text{e como (1) nos diz que } 13 = (1 \times 13).$$

$$6 = (1 \times 396) - (30 \times (1 \times 13)).$$

$$1 = (1 \times 13) - (2 \times 6) \dots \text{e como (2) nos diz que } 6 = (1 \times 396) - (30 \times 13).$$

$$1 = (1 \times 13) - 2 \times ((1 \times 396) - (30 \times 13)).$$

$$1 = (1 \times 13) - (2 \times 396) + (60 \times 13).$$

$$1 = (61 \times 13) - (2 \times 396).$$

O multiplicador de 13 é o inverso de 13  $(\text{mod } 396)$ . Neste caso, se encontrado o inverso de  $d$ , então  $d = 61$ . Portanto a chave privada  $(d, n) = (61, 437)$ .

Segundo Tkotz (2005), de posse das duas chaves, uma mensagem criptografada pode ser enviada. Esta mensagem precisa ser transformada num número para que o algoritmo possa ser aplicado. Para isto, os valores ASCII dos caracteres podem ser usados:

Mensagem: **Engenharia.**

**E =69, n =110, g =103, e =101, n =110, h =104, a =97, r =114, i =105, a =97,** ou seja, **m= 691101031011101049711410597.**

Tem-se a chaves públicas  $(e, n) = (13, 437)$ , e a chave privada  $(d, n) = (61, 437)$ . O próximo passo é dividir a mensagem **m**, em blocos:  $m_1 = 69; m_2 = 110; m_3 = 103; m_4 = 101; m_5 = 110; m_6 = 104; m_7 = 97; m_8 = 114; m_9 = 105; m_{10} = 97$ . Os blocos são submetidos a seguinte cifragem usando a chave pública, onde  $n$  é o módulo e  $e$  é o expoente:

$$c = m^e \pmod{n}.$$

$$\begin{aligned}
 c_1 &= 69^{13} \pmod{437} = 69. \\
 c_2 &= 110^{13} \pmod{437} = 48. \\
 c_3 &= 103^{13} \pmod{437} = 293. \\
 c_4 &= 101^{13} \pmod{437} = 403. \\
 c_5 &= 110^{13} \pmod{437} = 48. \\
 c_6 &= 104^{13} \pmod{437} = 61. \\
 c_8 &= 114^{13} \pmod{437} = 114. \\
 c_9 &= 105^{13} \pmod{437} = 146. \\
 c_{10} &= 97^{13} \pmod{437} = 136.
 \end{aligned}$$

A mensagem cifrada será 6948293403486136114146136. A decriptografia da mensagem é feita usando a mesma exponenciação, porém usando a chave de decifragem ou chave privada:

$$m = c^d \pmod{n}.$$

$$\begin{aligned}
 m1 &= 69^{61} \pmod{437} = 69. \\
 m2 &= 48^{61} \pmod{437} = 110. \\
 m3 &= 293^{61} \pmod{437} = 103. \\
 m4 &= 403^{61} \pmod{437} = 101. \\
 m5 &= 48^{61} \pmod{437} = 110. \\
 m6 &= 6^{61} \pmod{437} = 104. \\
 m7 &= 136^{61} \pmod{437} = 97. \\
 m8 &= 114^{61} \pmod{437} = 114. \\
 m9 &= 146^{61} \pmod{437} = 105. \\
 m10 &= 136^{61} \pmod{437} = 97.
 \end{aligned}$$

Com os resultados dos cálculos tem-se a mensagem decriptografada:

**691101031011101049711410597.**

Tem-se, assim, uma demonstração da complexidade do algoritmo RSA.

Mesmo com toda essa complexidade referida, atualmente o RSA é considerado um dos algoritmos de criptografia de chave pública mais usado em aplicações comerciais na internet, por exemplo, é utilizado nas mensagens de e-mails, em compras online e entre outras, de modo a garantir as empresas e usuários domésticos que seus compradores e funcionários, no primeiro caso, tenham segurança nas suas comunicações e transações e no segundo caso, ocorra a proteção de informações pessoais e compras seguras. Portanto, no atual momento, as transações comerciais são codificadas e decodificadas pela criptografia RSA, em que sua segurança está baseada na dificuldade de fatorar números inteiros grandes.

No apêndice desta dissertação encontrasse uma demonstração em Maple da criptografia RSA.

### 3.3 Criptografia de Curva Elíptica

Por sua necessidade de poder computacional menor assim como o tamanho das chaves a criptografia de curva elíptica tem sido aplicada em dispositivos móveis, embora o *NIST* informe que são necessários mais testes para o efetivo uso desta criptografia, ela pode ser encontrada em diversos dispositivos moveis. O princípio básico do problema desta criptografia é que quando um corpo envolve aritmética modular, a função inversa é intratável, e o tamanho da curva determina complexidade do problema. A criptografia baseada em *ECC* (*Elliptic Curve Cryptography*) mapeia o texto claro em pontos da curva previamente estabelecida.

Existem vários protocolos para implementação, deste podem ser destacados os seguintes, **ECDSA**, **ECDH**, **ECMQV** e o **ECDLP**. Estes algoritmos são baseados em hipóteses e protocolos pré-estabelecidos como protocolo de Diffie-Hellman (BLAKE SEROUSSI, 1999). Foram propostos em (KOBILITZ, 1987) e (MILLER, 1986) de maneira independente um do outro, os primeiros trabalhos com o uso de curvas elípticas em projetos de sistema criptográfico de chaves públicas, tendo como base um determinado agrupamento de curvas elípticas sobre um campo finito. Desde então várias pesquisas vêm sendo publicadas sobre a segurança e uma eficiente implementação de criptografia usando curvas elípticas.

### 3.3.1 Características das curvas elípticas.

As curvas elípticas são definidas sobre algum corpo, onde Corpo é um conjunto e mais duas operações com características especiais. O corpo dos reais é composto pelo conjunto dos reais mais as operações de adição e multiplicação. Os corpos utilizados na prática são os chamados “**corpos finitos**”, mas sera exemplificado com o corpo dos reais.

Curvas elípticas são estudadas e usadas em diversas áreas há muitos séculos, conforme (LENS-TRA, 1987), um exemplo disto foi sua utilização na prova do Último Teorema de Fermat e ainda existem aplicações em fatoração de números inteiros e, segundo GOLDWASSER (1999), em testes de primalidade. Uma das principais vantagens da ECC (*Criptografia por Curvas Elípticas*) é o seu tamanho pequeno de chave. Uma chave de 160 bits em ECC garante a mesma segurança que a chave de 1024 bits RSA como poderá se visto nesta dissertação.

A segurança do ECC depende da dificuldade do algoritmo da curva elíptica. Seja  $P$  e  $Q$ , dois pontos sobre uma curva elíptica tal que  $kP = Q$ , em que  $k$  é um escalar. Tendo-se  $P$  e  $Q$ , é computacionalmente inviável para se obter  $k$ , se for  $k$ , suficientemente grande.  $K$  é o logaritmo discreto de  $Q$  para a base  $P$ . Portanto, sua segurança está baseada no problema do logaritmo discreto. Daí a principal operação em ECC é ponto de multiplicação, ou seja, uma multiplicação  $k$  escalar com qualquer ponto  $P$  sobre a curva para se obter um outro ponto  $Q$  na curva.

Segundo BARBOSA (2003), ECDH (*Elliptic curve Diffie–Hellman*) é um protocolo de acordo de chave que permite que duas partes estabeleçam uma chave secreta compartilhada que pode ser utilizada para os algoritmos de chave privada. Ambas as partes trocam informação públicas um para o outro. Usando esses dados públicos e seus próprios dados privados para calcular o segredo compartilhado. Qualquer terceiro, que não tem acesso a chave privada de cada dispositivo, não será capaz de calcular o segredo partilhado entre os dois.

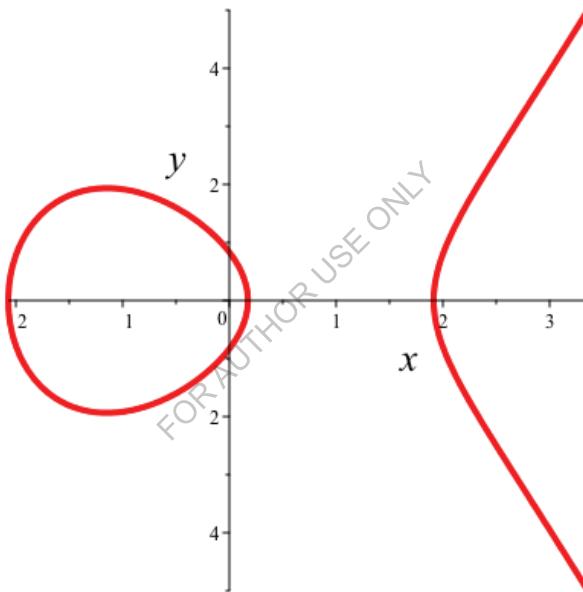
Para gerar um segredo compartilhado entre  $A$  e  $B$  usando *ECDH*, ambos têm que concordar com os parâmetros de domínio curva elíptica. Ambos ao final têm um par de chaves consistindo de uma chave privada  $d$ , um inteiro selecionado aleatoriamente inferior a  $n$ , onde  $n$  é a ordem da curva e uma chave pública. Tem-se  $q = d \times g$ , onde  $g$  é o ponto gerador, um parâmetro de domínio curva elíptica). Uma vez que é praticamente impossível encontrar a chave privada a partir da chave pública, não é possível obter o segredo compartilhado para um terceiro. A multiplicação é a principal operação na criptografia de curva elíptica. O inverso multiplicativo é uma operação dispendiosa em ambos os campos finitos, representando os pontos em projetiva. Sistemas de coordenadas podem eliminar a

necessidade de operação inversa multiplicativa além-ponto e ponto duplicação e com isso aumentar a eficiência da operação de multiplicação ponto.

Na figura 9, tem-se a representação de uma curva elíptica, que servirá como base para as explicações a seguir.

Esta curva foi gerada no aplicativo Maple e seu código encontrasse no apêndice desta dissertação.

Figura 9 – Exemplo de Curva Elíptica.



Fonte: Fonte: Maple 2016.

## 3.4 Protocolo de criptografia em curva elíptica

Segundo Dahab e López-Hernández (2007), protocolos numa definição direta são um conjunto de regras que torna possível a execução de um programa de modo eficiente e sem erros, assim, a importância do uso de protocolos na criptografia é sem precedentes. Protocolos de chaves criptográficas são desta forma ferramentas primordiais para o desenvolvimento de soluções de segurança que são clamadas no que se refere à segurança das comunicações e de circulação de dados e informações. A seguir, têm-se dispostos os principais protocolos usados em *ECC (Criptografia de Curva Elíptica)*.

### 3.4.1 Protocolo ECDSA

O ECDSA é uma variável do Algoritmo DSA utilizando curva Elíptica. O *Elliptic Curve Digital Signature Algorithm*, ou ECDSA é um protocolo de assinatura digital, que por sua vez é um algoritmo baseado no subgrupo proposto pelo *National Institute of Standards and Technology* (NIST, 2009). Está descrito nos seguintes documentos: ANSI X9.62, FIPS 186-2, IEEE 1363-2000 e ISO/IEC 15946-2.

### 3.4.2 Protocolo ECDH

O ECDH é baseado na variação do protocolo Diffie-Hellman. É conhecido como protocolo de chave criptográfica Diffie-Hellman de curva elíptica ECDH, o mesmo permite que dois usuários possam criar um contrato de segredo compartilhado, ou seja, possam entre si codificar e decodificar mensagens que estejam circulando entre eles.

### 3.4.3 Protocolo ECMQV

O ECMQV tem uma particularidade em que ambos os pontos presume chaves públicas estáticas. Encontra descrição nos seguintes documentos ANSI X9.63, IEEE 1363-2000, e ISO/IEC 15946-3.

### 3.4.4 Protocolo ECDLP

Este protocolo usa o modelo de multiplicação escalar. Este é o Problema de Logaritmo Discreto Sobre Curva Elíptica (*Elliptic Curve Discrete Logarithm Problem* - ECDLP, a segurança de criptossistemas baseados em curvas elípticas deste tipo de protocolo se baseia na dificuldade de

resolução do Problema de Logaritmo Discreto (*Discret Logarithm Problem - DLP*) sobre um grupo de curva elíptica.

### 3.4.5 ElGamal

Com o intuito de exemplificar é possível criar um modelo análogo ao ElGamal para um grupo de curva elípticas sobre corpos finitos, para este fim será considerado que Maria deseja enviar uma mensagem a Paulo e esta mensagem é um ponto da curva elíptica, onde o primo  $p$  e curva sobre  $Z_p$  são previamente acertados e seu compartilhamento se faz por uma canal de comunicação inseguro.

O método é descrito nas tabelas 3, 4 e 5:

Paulo inicia o processo e envia os dados a Maria.

Tabela 3 – ECC ElGamal Criptografia.

Paulo	Meio Público	Maria
	$p \gg$ Primo grande $E$ uma curva elíptica sobre $Z_p$ $P$ um ponto de $E(Z_p)$	
Escolhe $k \in Z$ secreto.		
Calcula $na = kP$		
Envia $na$ Para Maria		
	$na$ no meio público	

Fonte: Adaptado de Correia (2011).

Tabela 4 – ECC ElGamal Decriptografia.

Paulo	Meio Público	Maria
		Maria escolhe $k1 \in \mathbb{Z}$ e calcula $y = k1P$ $z = M + k1na$ Envia para Paulo a seguinte mensagens $(z, y)$
Recebe $(z, y)$	$(z, y)$ são públicos	
Para decifrar a mensa- gem é preciso calcular  $\begin{aligned} z &= nay \\ &= (M + ky - nakP) \\ &= M + k(naP) - naP \\ &= M \end{aligned}$		

Fonte: Adaptado de Correia (2011).

Na tabela 5 aplicam-se valores as variáveis, os cálculos podem ser comprovados no apêndice desta dissertação e para isso usou-se o aplicativo Maple.

Tabela 5 – ElGamal Decriptografia.

Paulo	Meio Público	Maria
	$p = 751, P = (0, 376)$ $E : y^2 = x^3 - x + 188$	
Chave secreta de Paulo  $k = 85$ Paulo Calcula  $na = kP = (671, 558)$ e então envia a Maria		
	na público	Recebe na Maria escolhe $k1 = 113$ e a mensagem $M = (82, 80)$ Então calcula $y = k1P = (34, 663)$ $z = M + k1na = (594, 179)$ Maria envia a Paulo a mensagem $(z, y)$
Recebe $(z, y)$ Paulo calcula  $z - nay =$ $(M + kna) - nakP$ $M + k(naP) - nakP$ $M = (82, 80)$	$(z, y)$ são públicos	

Fonte: Adaptado de Correia (2011).

### 3.5 Tamanho de chaves RSA e curva Elíptica

Com o avanço do poder computacional dos dispositivos, assim como o armazenamento é preciso ter atenção quanto ao tamanho das chaves usadas para criptografar os dados. Transmitir os dados de forma rápida e segura são requisitos fundamentais. Comparativamente as chaves usadas para criptografar usando **ECC** são muito menores que em outros tipos de criptografia.

O nível de segurança em sistemas está se tornando uma preocupação primordial como seria de esperar. A maioria dos especialistas recomendam que os sistemas atuais ofereçam ao menos 128 bits de segurança, mas o que isso realmente significa?

Isto não é a mesma coisa que comprimento da chave como muitos podem pensar. Segurança está na combinação do algoritmo específico e o seu tamanho da chave. Por exemplo, pensa-se geralmente que 128 bits de segurança pode ser conseguido com chaves de 128 bits AES, chaves de curva elíptica de 256 bits e chaves RSA 3072 bits. Se na implementação questões forem ignorados, então estes algoritmos com esses comprimentos de chave especificados geralmente terão o mesmo nível de segurança. Como pode ser visto na tabela 6 a criptografia RSA típica empregam atualmente 1024 ou 2048 bit, ainda que ambos sejam menos seguras do que AES-128.

Tabela 6 – Comparaçao de tamanho de chaves *AES, ECC e RSA*.

Simétrico	Elíptico	RSA
80	163	1024
128	256	3072
192	384	7680
256	512	15360

Fonte: Gupta et al. (2002).

### 3.6 Protocolo ECDH - Diffie Hellman

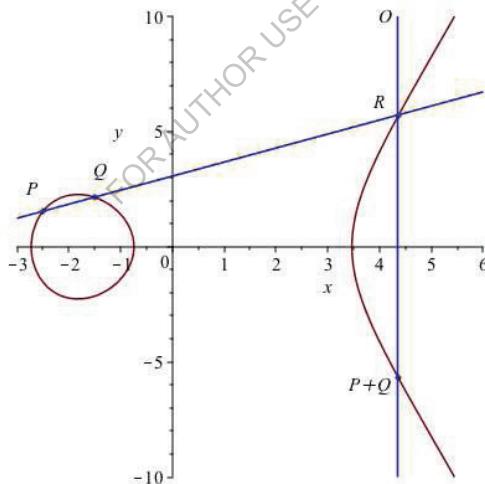
Como mencionado anteriormente a Criptografia de Curva Elíptica (ECC) é uma criptografia de chave pública. Na chave pública desta criptografia cada usuário ou o dispositivo que tomar parte na comunicação têm, geralmente um par de chaves, uma chave pública e uma chave privada, e um conjunto de operações associadas as chaves para fazer as operações criptográficas. Somente o usuário em particular sabe a sua chave privada. Considerando que a chave pública é distribuída a todos os utilizadores que participam na comunicação. Algum algoritmo de chave pública pode exigir um

conjunto de constantes predefinidas a ser conhecidos por todos os dispositivos, que fazem parte na comunicação. Segundo Klima, Sigmon e Stitzinger (2006), "Parâmetros de domínio em ECC é um exemplo de tais constantes". Criptografia de chave pública, ao contrário de criptografia de chave privada, não requer nenhum segredo compartilhado entre as partes que se comunicam, mas é muito mais lento do que a criptografia de chave privada. As operações matemáticas da ECC são definidas sobre a curva elíptica e pode ser vista na equação a seguir:

$$y^2 + axy + by = x^3 + cx^2 + dx + e.$$

Cada valor de 'a' e 'b' dá uma curva elíptica diferente. Todos os pontos ( $x, y$ ) que satisfaçam a equação acima, mais um ponto no infinito fica na elíptica curva. A chave pública é um ponto da curva e a chave privada é um número aleatório. A chave pública é obtida multiplicando a chave privada com o ponto gerador  $G$  na curva, os parâmetros da curva de 'a' e 'b', juntamente com mais alguns constantes constituí o parâmetro do domínio de ECC, isto pode ser visto na figura 10.

Figura 10 – Curva Elíptica.



Fonte: Fonte: Maple 2016.

### 3.7 Comparação entre RSA e ECC.

O modelo RSA é o modelo usado atualmente para criptografia em chave pública e usa como esquema a fatoração de inteiros como problema matemático, sendo que sua resistência a criptoanálise reside na dificuldade da fatoração de números inteiros grandes. O tamanho desses números torna o ataque por força bruta inviável (HAZAY et al., 2007). Na criptografia baseada em curvas elíptica tem-se uma redução considerável no processamento por não envolver problemas complexos de fatoração e suas chaves serem menores, esta característica fazem com que sua aplicação em dispositivos com menor poder computacional a torne mais eficiente. Desta forma, se faz necessária a aferição dos tamanhos das chaves, que já foi feita, e dos níveis de segurança que são ofertados pelos diferentes protocolos, focalizando nos protocolos de curva elíptica (ECC) e no protocolo (RSA).

A Sun, através do SSL, mostrou que a criptografia elíptica (ECC) realmente é tão mais eficiente que a criptografia com RSA quanto maior for a segurança necessária, (GUPTA et al., 2002). Como pode ser visto na figura 7.

Tabela 7 – Eficiência ECC.

	RSA enc,ver	RSA decry,sign	ECDSA ver	ECDSA sign	ECDH op
Ultra - 80	1,7	31,1	13,0	6,8	6,1
Yopy	10,8	188,7	46,5	24,5	22,9

Fonte: Gupta et al. (2002).

Tempos (ms) de execução das operações criptográficas básicas, para RSA 1024 bits e ECC 163 bits na tabela 8 e RSA 2048 bits e ECC 193 bits na tabela 8.

Tabela 8 – RSA e ECC

	RSA enc,ver	RSA decry,sign	ECDSA ver	ECDSA sign	ECDH op
Ultra - 80	6,1,	205,5	18,1	9,2	8,7
Yopy	39,1	1273,8	76,6	39,0	37,7

Fonte: Gupta et al. (2002).

Na 5 pode ser visto uma comparação apresentada por NIST (National Institute of Standards and Technology, 2007) acerca dos níveis de segurança entre os diferentes algoritmos que podem ser

usados na atividade criptográfica, a tabela em referência mostra uma comparação do nível de segurança provido pelos algoritmos que são aprovados pelo instituto NIST.

Tabela 9 – Nível de segurança.

Bits de Segurança	Simétrico	DSA	RSA	ECDSA
80	2TDEA		18,1	9,2
Yopy	39,1	1273,8	76,6	39,0

Fonte: Gupta et al. (2002).

### 3.8 Escolha da Curva Elíptica.

Uma analise teórica foi realizada no capítulo anterior, porém existem muitas curvas que podem ser usadas com a finalidade de manter os dados seguros. Neste capítulo será abordado quais curvas podem ser recomendadas para utilização no sistema do *playout* em nuvem. Segundo Ladeira (2016), a **NSA(National Security Agency)**, recomenda para o governo americano dez escolhas para corpos finitos, sendo cinco corpos primos ( $F_p$ ) e cinco para corpos binários ( $F_2^m$ ) (BROWN et al., 2001). Os corpos são listados a seguir com seus respectivos polinômios, onde  $P$  é corpo primo e  $B$  é corpo binário:

$$(P - 192)$$

$$F_{192} \rightarrow p = 2^{192} - 2^{64} - 1;$$

$$(P - 224)$$

$$F_{224} \rightarrow p = 2^{224} - 2^{96} + 1;$$

$$(P - 256)$$

$$F_{256} \rightarrow p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1;$$

$$(P - 384)$$

$$F_{384} \rightarrow p = 2^{384} - 2^{128} - 2^{96} + 2^{32} - 1;$$

$$(P - 521)$$

$$F_{521} \rightarrow p = 2^{521} - 1;$$

$$(B - 163)$$

$$F_2^{163} \rightarrow f(x) = x^{163} + x^7 + x^6 + x^3 + 1;$$

(B - 233)

$$F_2^{233} \rightarrow f(x) = x^{233} + x^{74} + 1;$$

(B - 283)

$$F_2^{283} \rightarrow f(x) = x^{283} + x^{12} + x^7 + x^5 + 1;$$

(B - 409)

$$F_2^{409} \rightarrow f(x) = x^{409} + x^{87} + 1;$$

(B - 571)

$$F_2^{571} \rightarrow f(x) = x^{571} + x^{10} + x^5 + x^2 + 1.$$

Segundo Koblitz (1987), os corpos primos e binários foram determinados sob a perspectiva do seu desempenho e a simplificação da aritmética utilizada. Existem curvas genéricas binárias, também conhecidas como curvas de *Koblitz*, porém não serão abordadas nesta dissertação. As curvas listadas foram geradas de forma **pseudo-aleatória** e descartou-se as que fossem comprovadamente resistentes a ataques conhecidos. O gerador de bits pseudo-aleatório e sua semente não forma escolhidos de forma transparente, (LADEIRA, 2016) e com isso tem gerado resistência a sua aplicação pela comunidade técnica, que suspeita haver a possibilidade das sementes escolhidas terem vulnerabilidades conhecidas pela NSA , mas desconhecida pelo público em geral.

Desta forma as curvas elípticas tem recebido grande atenção pela área científica e também pela indústria, por terem seus métodos de geração mais transparentes e alto desempenho na implementação e protegidas contra ataques de canal lateral de tempo, segundo Silva (2015), ataques de canais laterais são baseados em tempo e permitem a um adversário monitorar pequenas flutuações no tempo de execução do algoritmo criptográfico. Essas variações são devidas aos desvios condicionais, otimizações no nível de instruções, desempenho da hierarquia de memória ou latência de comunicação, porém não terão um maior aprofundamento nesta dissertação.

## 4 APLICAÇÃO

Para o modelo proposto como sistema de compartilhamento de conteúdo, algumas premissas foram estabelecidas, conforme citadas nos capítulos iniciais desta dissertação:

Sistema em nuvem – AWS - Dupla Autenticação.

Sistema de controle de colaboração - *MAM*

Usuário cadastrado e autenticado.

*Upload e download* de arquivos

Metadados

Compartilhamento de conteúdo.

Transcodificação de vídeo.

Visualizar conteúdo.

Banco com TDE e auditoria.

Storage em nuvem - Playout

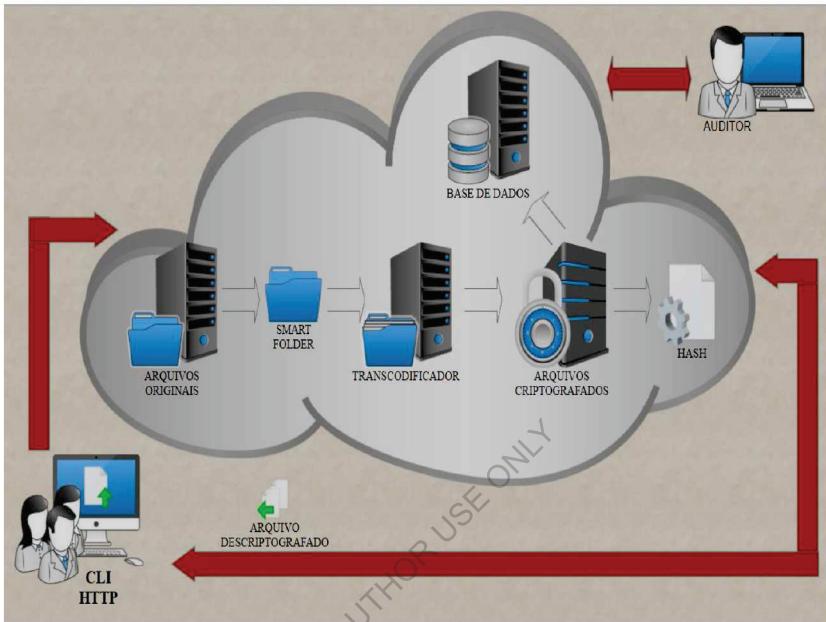
Aplicação de ECC

Estas premissas servem como referência para atender ao ambiente de *broadcast* em sua complexidade e suas característica no que tange a segurança.

O sistema deve ter como linha norteadora a resiliência das operações, onde é possível estabelecer a recuperação das atividades em tempo reduzido, comparado a sistemas discretos similares.

A figura 11, representa o ambiente do sistema e sua aplicação, onde pode-se destacar a utilização de mecanismo de segurança e de alta disponibilidade do sistema, além de uma interface amigável para sua utilização.

Figura 11 – Aplicação.



Fonte: Elaborada pelo autor.

Vale destacar que a operação de transcodificação deve ser transparente para o usuário do sistema, visto que os *presets*, serão previamente configurados pelos especialistas, a exceção a esta condição existe apenas no ambiente de pré-visualização que será demonstrado no capítulo acerca do MAM.

A escolha do provedor de serviço foi feita com base em uma pesquisa dentre vários outros provedores, e a AWS se mostrou com maior aderência ao ambiente proposto, assim como o conhecimento da equipe envolvida que adiante será responsável pelo desenvolvimento, implantação e manutenção do sistema.

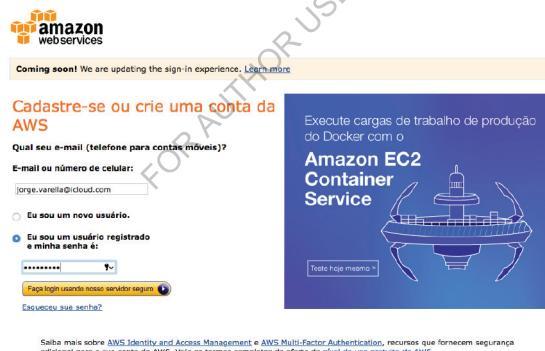
## 4.1 Ambiente em Nuvem

Para compor a infraestrutura que atendesse as especificações dos itens dos sistemas envolvidos, como o gerenciador de conteúdo (MAM), o banco de dados e transcoder de vídeo, se fez necessário instanciar máquinas no ambiente em nuvem da AWS, com as configurações descritas a seguir. Segundo Amazon (2017), o ambiente escolhido Amazon EC2, (*Amazon Elastic Computer Cloud*) pode ser definido como um serviço Web que fornece capacidade computacional segura e com a possibilidade de redimensionamento na nuvem. Estas características são de extrema importância por conta de períodos onde se alternam altas demandas e outros de ociosidade na utilização da transcodificação de vídeo, além da segurança do conteúdo ser uma das características do sistema.

Conforme as premissas do sistema destacadas no capítulo anterior, este ambiente deverá fornecer dupla autenticação aos administradores do sistema conforme pode ser evidenciado a seguir.

Na figura 12, pode ser visto o primeiro nível de autenticação no ambiente da AWS. Na primeira autenticação é necessário fornecer apenas usuário e senha pré-cadastrados.

Figura 12 – Primeira Autenticação.



Fonte: Fonte: Amazon (2017).

O nível de segurança em nuvem onde o acesso é controlado apenas com usuário e senha deve ser considerado baixo, visto que o extravio destas informações compromete todo o sistema, porém a

autenticação dupla é uma camada extra de segurança, pois além de fornecer usuário e senha em sua autenticação conforme visto na figura 12 é obrigatório fornecer um código de seis dígitos gerado de forma pseudoaleatório por um *token* que pode ser instalado em um dispositivo móvel.

Na figura 13 é possível observar a solicitação feita pelo sistema de um código de autenticação.

Figura 13 – Segunda Autenticação.



Fonte: Fonte: Amazom (2017).

A figura 14, exibe o código gerado pelo *token* que permitirá a utilização dos sistema AWS e que será solicitado todas as vezes o que o usuário efetuar seu login, porém vale salientar que a dupla autenticação não é por padrão habilitado, sendo necessário configurar esta opção no ambiente de configuração AWS.

Figura 14 – Código Dupla Autenticação.



Fonte: Fonte: Print Screen de Tela

As máquinas estanciadas no ambiente AWS, foram criadas de acordo com as especificações dos sistemas que nelas iriam ser instalados e que não impactassem no cumprimento das premissas propostas.

A primeira máquina instanciada foi configurada com dois processadores e 16 GB de RAM, um disco SSD de 45 TB que foi usado para instalação do sistema operacional Ubuntu 16 e dos sistema de gerenciamento de conteúdo.

A segunda máquina instanciada possui dois processadores e 32 GB de RAM, disco SSD de 45 TB e foi usada para instalação do Banco de dados sob um sistema operacional Ubuntu 16.

A terceira instancia construída foi uma máquina com quatro processadores, 64Gb de RAM e disco SSD de 120TB, esta máquina foi instanciada com estas características por conta de sua utilização demandar grandes recursos computacionais e espaço em disco. cujo sistema operacional instalado foi o windows server 2016.

A figura 15, mostra as instâncias no ambiente AWS utilizadas na criação do sistemas proposto. Por uma questão de custo as máquinas somente formam iniciadas durante sua efetiva utilização, é possível ainda observar varias outras possibilidades de configuração como os volumes que podem ser agregados às instâncias, porém não será dado destaque as suas funcionalidades nesta dissertação.

Para cada uma das máquinas foram criadas duas interfaces de rede. Esta característica se deve ao fato da necessidade de redundância do acesso e a possibilidade de balanceamento de carga durante os testes com o sistema.

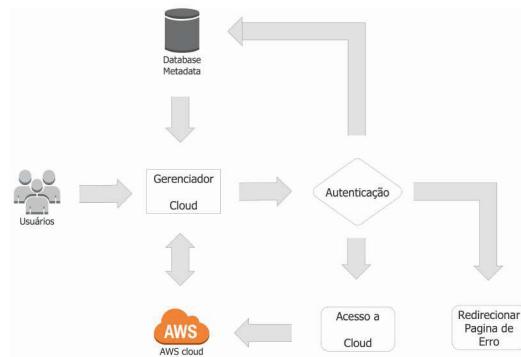
Figura 15 – Tela Amazon.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
Ubuntu 16 R...	i-06044ff5b32f62dd5	t2.micro	sa-east-1c	running	2/2 checks ...	None	ec2-18-231-44-123
MySQL	i-08b857ff04ebd248	t2.micro	sa-east-1a	running	2/2 checks ...	None	ec2-54-207-76-244
LowCost-env	i-0a4a8e05604ad3...	t1.micro	sa-east-1a	running	2/2 checks ...	None	ec2-52-67-254-230
<b>Transcoder</b>	<b>i-0c6853e277c422e8f</b>	<b>t2.micro</b>	<b>sa-east-1c</b>	<b>stopped</b>			

Fonte: Fonte: Amazon (2017)

A figura 16 ilustra o ambiente de acesso no estado da arte, onde destaca-se o gerenciamento de acesso feito na infraestrutura AWS, banco de dados que será detalhado em um capítulo dedicado a isto.

Figura 16 – Acesso ao sistema.



Fonte: Fonte: Jain, Jain e Kapil () .

## 4.2 MAM - Media Asset Management

O gerenciador de conteúdo, (MAM) foi selecionado dentre vários possíveis que atenderiam as premissas estabelecidas. Dentre os sistemas testados podem ser destacados os seguintes:

Dalet Galaxy - Dalet.

Vsn Explorer MAM -VSN.

Metus MAM - Metus.

Razuna MAM - Razuna.

O sistema Razuna MAM , foi o selecionado por esta em acordo com os seguintes fatores:

Poder ser executado em nuvem.

Cadastramento de usuários e autenticação em banco próprio ou com integração com AD.

*Upload e Download* de arquivos sem restrição de tamanho.

Entrada de metadados e possível incorporação de dicionário controlado.

Transcodificação de vídeo em vários formatos e com isso o usuário é quem define o modo de

pré-visualização do conteúdo.

Segundo Razuna (2017), o Razuna MAM é um sistema de colaboração de conteúdo de código aberto que permite que o usuário, em um ambiente centralizado de alta disponibilidade, seja capaz de tratar diversos tipos de arquivos, como por exemplo arquivos de vídeo, arquivos de áudio, documentos em vários formatos, tendo ainda controle sobre o compartilhamento deste conteúdo.

Na figura 17 é possível verificar as informações do formulário de cadastro de usuários, incluindo também a data de expiração de conta, funcionalidade extremamente pertinente para a área de *broadcast* onde a existência de programas ou séries com tempo pré-determinado de produção é uma constante. Conforme mencionado anteriormente a licença de uso do sistema é uma AGPL (Afferro General Public License) que segundo GNU (2017) é uma licença gratuita para software e outros tipos de trabalho, especificamente concebido para garantir cooperação com as comunidades e desenvolvedores.

Figura 17 – Razuna Cadastro.

The screenshot shows the Razuna MAM web interface. On the left, there's a sidebar with 'System' and 'Users / Groups' sections. The main area has a title 'Add User' with tabs for 'Add User', 'Groups', and 'Tenants/Hosts'. The 'Add User' tab is active. It contains fields for 'Username / eMail', 'Password', 'Confirm Password', and various contact details like 'First Name', 'Last Name', 'Salutation', 'Company', 'Telephone', 'Fax', and 'Mobile/Cell'. There's also a note about expiration dates. To the right, a table lists existing users with columns for 'eMail', 'Tenant Access', and icons for edit and delete. At the bottom, there's a footer with links to Razuna 1.9, AGPL license, and Razuna documentation.

eMail	Tenant Access
ri@uco.inf	Demo
jorge.varella@icloud.com	Demo
sp@uco.inf	Demo

Fonte: Razuna (2017).

O *upload* e *download* de arquivos são tarefas que devem ser executadas pelos usuários do sistema independentemente do navegador que estiver usando, esta característica deve ser respeitada para tornar o sistema compatível com os diversos navegadores existentes. Embora sabendo que respeitando o W3C (*World Wide Web Consortium*) o sistema seria compatível, foram realizados testes com os seguintes navegadores e todos foram aprovados:

Internet Explorer - Todos a partir da versão 9.

Chrome - Ultima versão.

FireFox - Ultima versão.

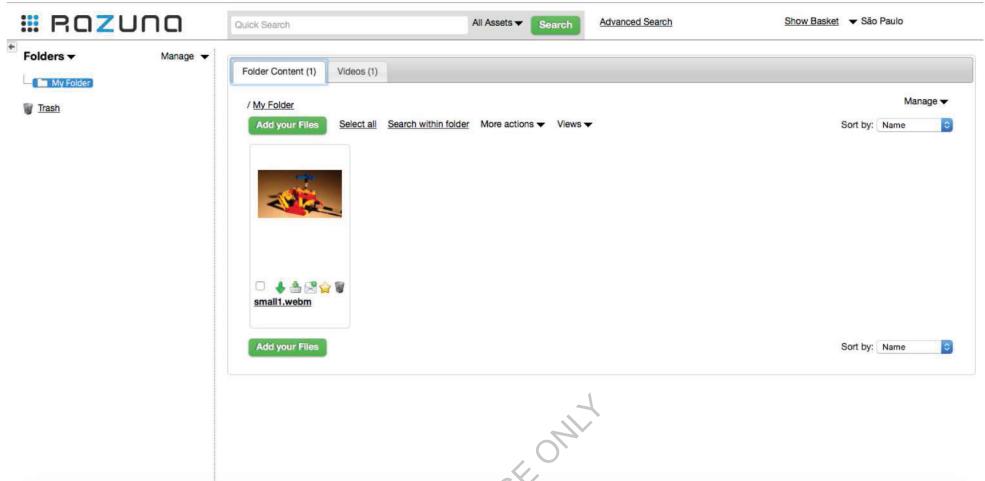
Opera - Ultima versão

Safari - Versão 11.

A figura 18 , mostra a tela do sistema em que é possível fazer *upload* e *download* de arquivos no sistema, quando esta operação é executada o arquivo é armazenado em local pré-determinado e que será tema do capítulo armazenamento.

Com o objetivo de tornar o ambiente mais agradável ao uso, os arquivos são separados por categoria, ou seja, uma área específica para arquivos de audio, documentos e arquivos de vídeo como pode ser visto na figura 18.

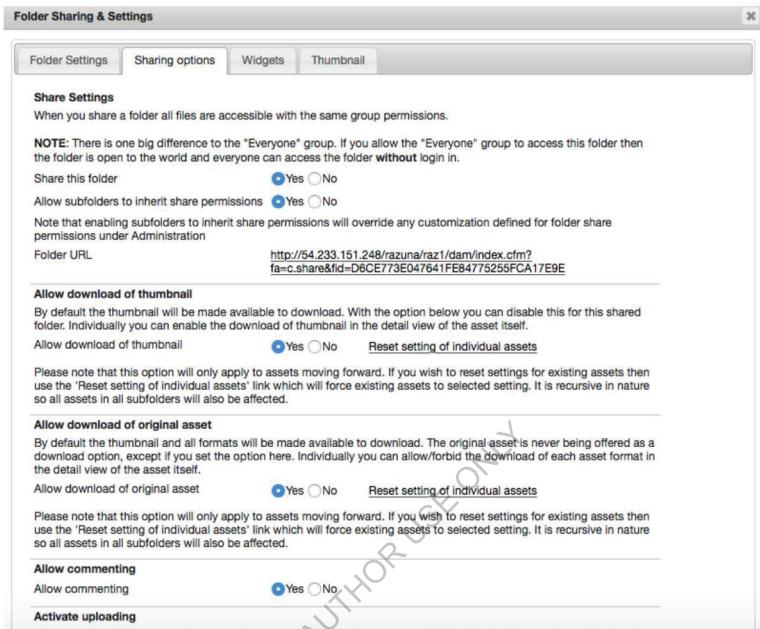
Figura 18 – Razuna Upload.



Fonte: Razuna (2017).

Para se realizar o *download* do material é necessário além do login no sistema, a permissão do usuário que postou o material, para que o mesmo possa ser baixado. O sistema permite uma série de configurações de compartilhamento que podem ser vistas na figura 19.

Figura 19 – Razuna Share.

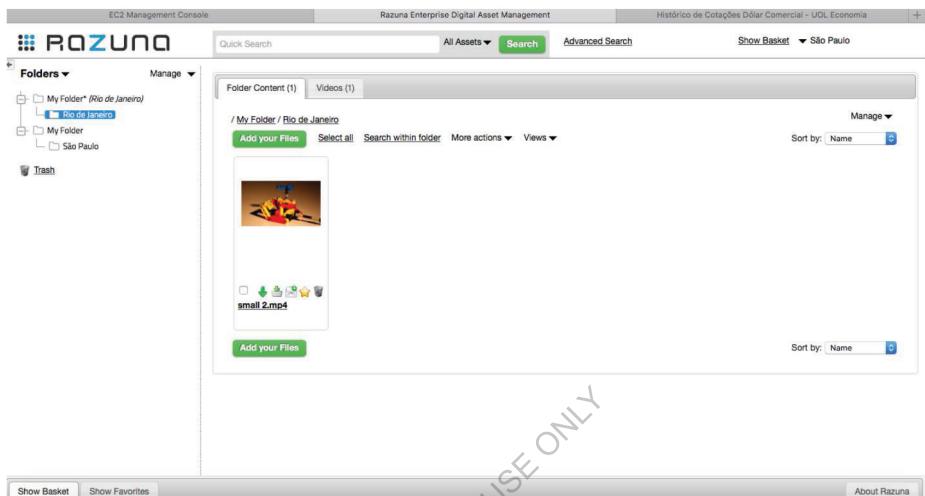


Fonte: Razuna (2017).

Dentre as configurações vale destacar a permissão de *download* de conteúdo original e o *link* para postagem do material convertido ou original no site do usuário que tenha permissão de acesso.

Na figura 20, é possível verificar o compartilhamento de pastas entre os usuários. Esta facilidade busca além da integração entre os diversos usuários o melhor aproveitamento de cada produção local, visto que o custo em questão para preenchimento da grade de programação passa a ser "rateado" entre os diversos membros.

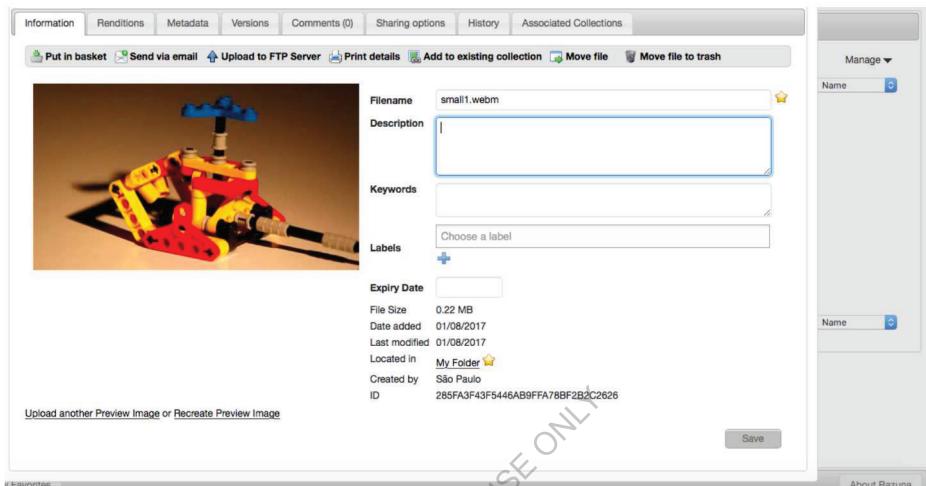
Figura 20 – Razuna Share Pastas.



Fonte: Razuna (2017).

Dentre as premissas do sistema encontra-se uma que é pertinente à busca precisa do arquivo compartilhado, a premissa em questão é a existência de campos de metadados referentes aos arquivos no momento em que o usuário está fazendo o *upload* do mesmo. Quanto maior for a precisão dos dados, mais eficaz será a busca posterior, caso o arquivo seja compartilhado. Os campos de metadados podem ser vistos na figura 21. Devem ser destacados os campos *ID* e *Keywords* como campos de identificação única de conteúdo e busca rápida, respectivamente.

Figura 21 – Metadados.



Fonte: Razuna (2017).

A existência de uma grande quantidade de *codecs* de vídeo no mercado de *broadcast* exige que o sistema a ser ofertado permita a transcodificação de conteúdo sem que seja exigido do usuário desembolso na compra destes *codecs*. O sistema permite em seu ambiente transcodificar o conteúdo original para diversos formatos, como pode ser visto na figura 22.

A aplicação usa na transcodificação o pacote *ffmpeg*, que segundo FFMPEG (2017) é um solução em código aberto para gravação, conversão e transmissão de áudio e vídeo, capaz de converter vídeos em formatos de alta resolução fazendo uso otimizado dos recursos computacionais. Os vários formatos possíveis de conversão podem ser vistos na figura 22.

Dentre os vários formatos disponíveis para transcodificação de vídeo devem ser destacados os formatos *avi*, *mpeg* e *mxif*, que são usados na quase totalidade dos *playout* (exibidores de conteúdo).

Existe ainda no ambiente de transcodificação a possibilidade de conversão de arquivos de áudio

e imagem como *photoshop* da Adobe, estas conversões podem ser úteis para futura publicação do conteúdo em sites, desde que não exista restrição para tal.

Figura 22 – Razuna Codecs.

Extension	File Type	Mime Content	Mime Sub-Content
3fr	Image	image	3fr
3go	Video	video	3gpp
aff	Audio		
alt	Audio		
ai	Image	application	photoshop
aif	Audio	audio	x-aiff
aifc	Audio	audio	x-aifc
aiff	Audio	audio	x-aiff
ari	Image	image	ari
arw	Image	image	arw
au	Audio	audio	basic
avi	Video	video	avi

Fonte: Razuna (2017).

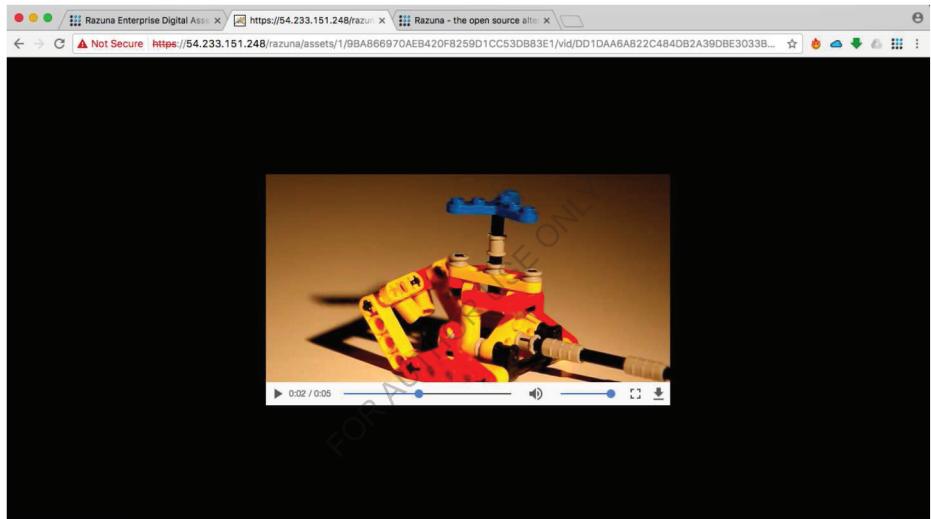
Uma outra premissa que o sistema do MAM deve respeitar é a pré-visualização do conteúdo, ou seja, com o arquivo disponibilizado, os usuários devem ter a possibilidade de visualizá-lo sem ter que baixar o mesmo para sua máquina. As considerações levantadas sobre metadados devem ser levadas em consideração neste momento, visto que logo após a busca, o usuário se depara com vários arquivos listados. Neste momento visualizar o material no próprio navegador torna a tarefa muito mais prática e eficiente, levando em consideração que o usuário irá fazer o *download* somente quando necessário e do que realmente deseja, fazendo um decupagem (escolha) dos itens disponíveis listados na busca.

A figura 23 mostra o *preview* de um vídeo no navegador do usuário, podendo este inclusive fazer o download sem necessidade de voltar na tela de busca. Uma das facilidades desta pré-visualização é a exibição do conteúdo em tela cheia, isto contribui de forma significativa no que tange

a primeira avaliação do material, visto que no *upload* de material não há restrição quanto a qualidade técnica do mesmo.

Esta pré-visualização deve ser usada também para a escolha do formato de transcodificação dos arquivos, pois a possibilidade de fazer uma avaliação, mesmo que subjetiva, irá auxiliar na tomada de decisão por parte do usuário.

Figura 23 – Razuna Preview.



Fonte: Razuna (2017).

### 4.3 Armazenamento

O local de armazenamento de conteúdo de acordo com a premissa do sistema deverá ser no ambiente em nuvem, contudo a escolha levou em consideração o custo deste armazenamento, a

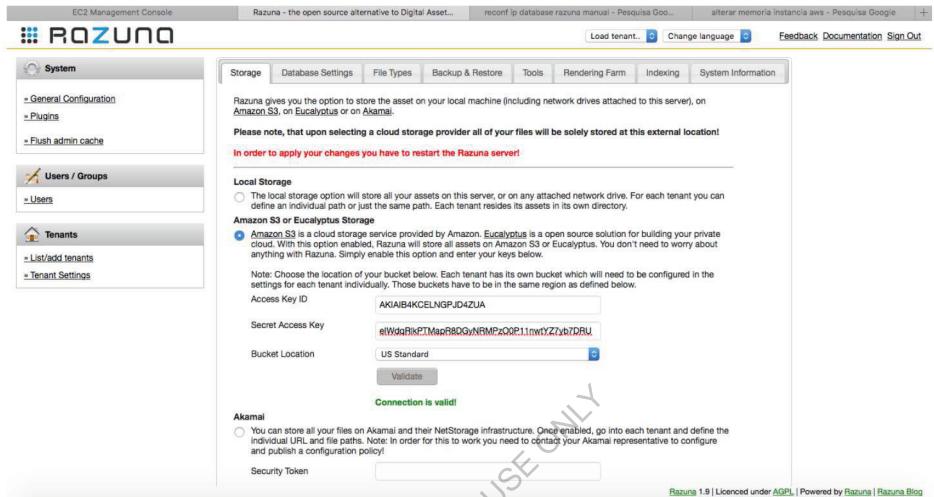
operacionalidade, a integração com o sistema MAM e a latência do material solicitado, pois isto impacta diretamente no sistema de exibição.

O sistema de armazenamento S3 da AWS foi o que apresentou melhor aderência ao sistema de gerenciamento de conteúdo, além de possuir um *datacenter* na cidade de São Paulo. A integração com o sistema de gerenciamento de conteúdo pode ser vista na figura 24. O sistema MAM Razuna permite a integração com o sistema de armazenamento S3 da AWS sem a necessidade de configuração *outside do sistema*.

Com a possibilidade de escolha do local de armazenamento foi necessário verificar qual local teria a menor latência na transferência de material. Nos testes realizados a menor latência foi apresentada no datacenter de São Paulo.

Foram realizados testes em outros dois serviços de armazenamento em nuvem, o *Eucalyptus Storage* e *Akamai*. Os dois ambientes não apresentaram problemas de integração, porém no quesito latência obtiveram menor eficiência comparados ao S3 da AWS. Estes testes foram empíricos, não houve disponibilidade de equipamento capaz de fazer medidas destas latências.

Figura 24 – Razuna S3 AWS.



Fonte: Razuna (2017).

Com a premissa do armazenamento sendo cumprida, o passo seguinte foi estabelecer o *backup* do sistema. O ambiente de *broadcast* pode ser traduzido em uma palavra "**resiliência**" que segundo UFGRS (2017), é a capacidade de manter níveis aceitáveis de operação frente a anomalias.

Na configuração realizada durante o período de testes não houve possibilidade de estimar com precisão o tempo de recuperação do sistema, pois não há uma métrica estabelecida por Megabytes armazenados. Esta métrica será estabelecida quando for populado o banco de dados e uma quantidade de material acima de 10 horas de vídeo armazenados pelo sistema.

Vale destacar que a contratação do serviço S3 da AWS deve ser realizado com base no números de horas de programação a serem armazenas para suprir a demanda dos participantes. O custo desta contração deverá ser rateado entre as partes de forma igualitária mensalmente.

As configurações de *backup* podem ser vistas na figura 25, assim como o *setup* de recuperação.

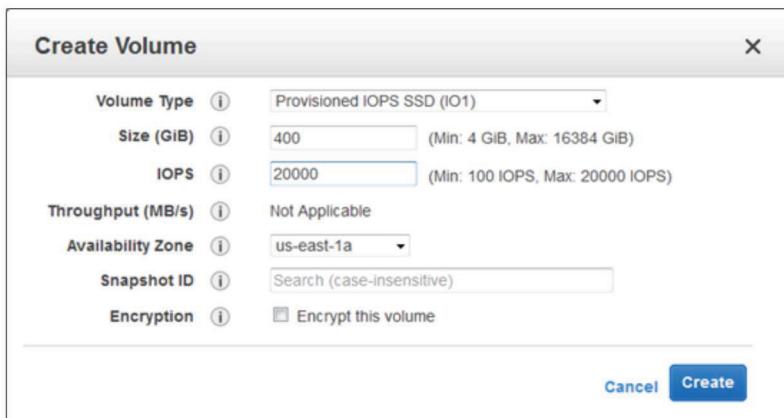
Figura 25 – Razuna Backup.

The screenshot shows the Razuna Management Console interface. The top navigation bar includes links for EC2 Management Console, Razuna - the open source alternative to Digital Asset..., alterar memoria instancia aws - Pesquisa Google, and other options like Load tenant, Change language, Feedback, Documentation, and Sign Out. The main menu on the left has sections for System (General Configuration, Plugins, Flush admin.cache), Users / Groups (List/add tenants, Users), and Tenants (List/add tenants, Tenant Settings). The central content area is titled 'Backup & Restore' and contains several sections: a general note about backing up and restoring the server, a 'Scheduled Backup' section with a 'Once Daily' option selected, a 'Save Schedule' button, a 'Restore Razuna Data' section with a note about restoring from the database, and a table for managing backups with columns for Backup Date, Restore, and Remove. At the bottom right, there's a footer with links to Razuna 1.9, AGPL, Powered by Razuna, and Razuna Blog.

Fonte: Razuna (2017).

A criação de um volume pode ser visto na figura 26, destacando a local disponível para criação do mesmo.

Figura 26 – S3 AWS.



Fonte: Razuna (2017).

#### 4.4 Transcoder FFTrans

Transcodificar o conteúdo ingestado pelos usuários é a garantia de que embora cada usuário tenha seu próprio *codec* de exibição, os vídeos e outros materiais, como áudio por exemplo, que podem ser entregues em um formato específico. Caso tenha-se um usuário que colabora ingestando um vídeo no formato *Mpeg 2*, pois seu sistema de exibição está preparado para este formato, com o sistema de transcodificação este mesmo material pode ser entregue a outro usuário em *MXF*, sem intervenção do usuário nesta operação.

O sistema de transcodificação usado foi o *FFSTrans*, software livre capaz de transcodificar para formatos *broadcast*. Este sistema é executado em ambiente *Windows* e para isto foi instanciada uma máquina na *Amazon*, conforme descrito no capítulo ambiente em nuvem.

O usuário necessita informar qual o seu formato de vídeo padrão e uma configuração é feita no aplicativo *FFTrans*. Um mapeamento é feito na estação de trabalho onde o exibidor será executado.

O processo de transcodificação segue os seguintes passos:

Usuário A faz o ingestão do material - *upload*.

O material é depositado em uma pasta no S3 AWS - Vídeo Original.

O aplicativo FFtrans identifica um novo material e faz a transcodificação e deposita o material na pasta de Vídeos Transcodificados.

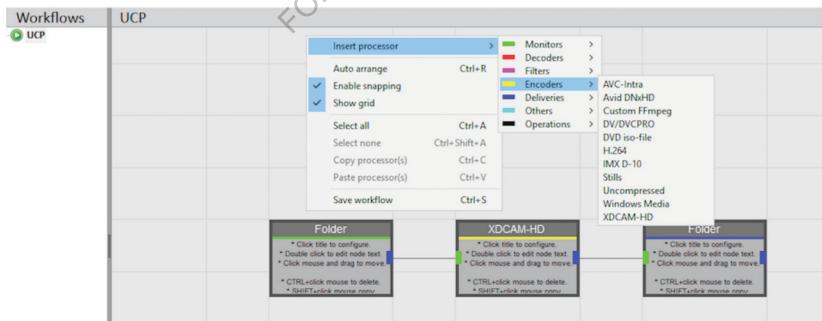
O usuário B tem acesso a pasta de vídeo transcodificados no seu formato de exibição.

Este fluxo pode ser visto na figura 27, assim como os vários formatos de vídeo com Avid, IMX e XDCAM, dentre vários outros.

Conforme mencionado anteriormente este serviço de transcodificação exige grande poder computacional em determinados momentos, alternados por períodos de ociosidade. A redução de custo em não manter em cada usuário do sistema uma máquina com grande poder computacional, o custo de manutenção e necessidade de atualização constante foram fatores determinantes para a utilização do ambiente em nuvem para este serviço.

O FFTrans, aplicativo para transcodificação usa a biblioteca FFMPEG *Broadcast* para a conversão de vídeo. Esta biblioteca difere da versão comumemente usada por conter diversos formatos profissionais de vídeo, conforme evidenciado na figura 27. Embora existam soluções comerciais com melhor desempenho, seu custo de utilização é de aproximadamente 10.000 dólares por processador, o que tornaria o projeto inexplorável.

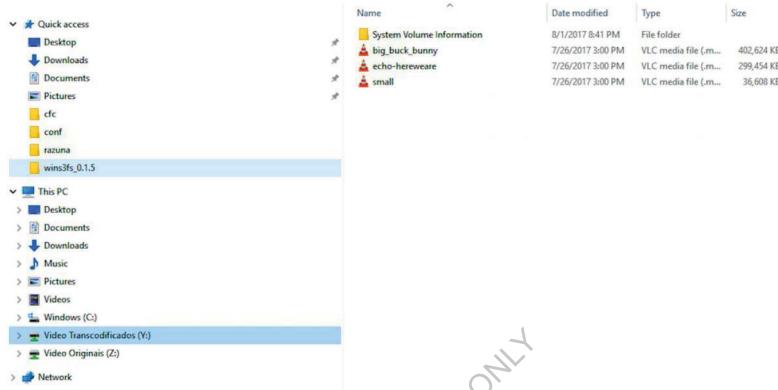
Figura 27 – Trasncoder FFTrans.



Fonte: Print de tela.

O mapeamento das pastas, conforme mencionado anteriormente, pode ser visto na figura 28, no ambiente *Windows*.

Figura 28 – Pastas Mapeadas Windows.



Fonte: Print Screen de tela.

## 4.5 Playout

O sistema de exibição (*Playout*) em nuvem é o sistema responsável por encadear a sequência de vídeos a serem exibidos por diferentes meios como: transmissão satelital, *stream* via internet, transmissão via cabo e transmissão digital terrestre. Este exibidor deve permitir importação do sistema de *playlist* (Relação de vídeos encadeados), possuir pré-visualização do material, deve conter o *time code* do material em exibição e um contador de tempo total do *playlist*.

Com esses pré-requisitos estabelecidos é possível encontrar no mercado de *broadcast* uma grande quantidade de soluções, assim sendo não foi imposto um modelo específico, ou seja, cada participante tem condições de escolher a solução que melhor se adequa a sua estrutura física e financeira.

O sistema de *playout* usado nos testes foi o *JustPlay* da empresa *Toolonair*. O *JustPlay* segundo a ToolonAir (2017) é um sistema de reprodução de vídeo de transmissão automática que pode

reproduzir vários formatos de vídeo com resoluções com o SD (*standard-definition* ao 4K, que é quatro vezes a alta resolução e que tem o *aspect ratio* de 1920 por 1080, linhas e colunas respectivamente, embora ainda não exista transmissão em 4K no Brasil o exibidor esta preparado para esta operação.

Na figura 29 pode-se verificar as condições estabelecidas para o sistema de exibição, a saber:

Pré-visualização do conteúdo a ser exibido.

Visualização do material em exibição.

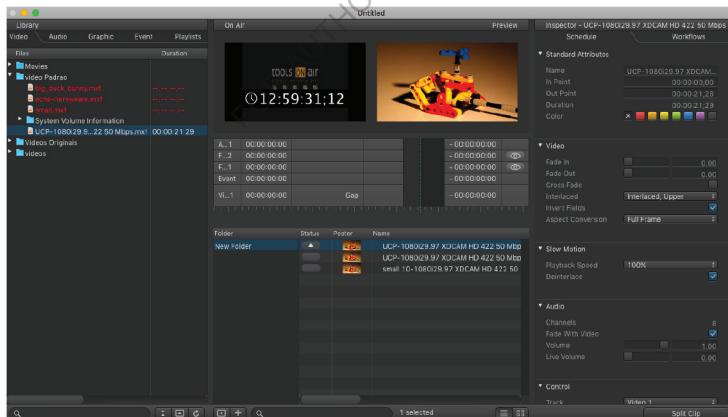
*Time code* de exibição.

*Time code* total do *playlist*.

Este sistema de exibição conta ainda com a possibilidade de alteração da sequência dos vídeos a serem exibidos, esta função permite aos operadores afinar o tempo total da grade de programação. Este ajuste é de grande valia em operações comerciais onde determinados vídeos, (peças) devem respeitar horário de exibição.

Não obstante, é preciso evidenciar o mapeamento dos volumes em nuvem e o sincronismo para o *storage* local do sistema.

Figura 29 – Playout.



Fonte: ToolonAir (2017).

## 4.6 Banco de dados

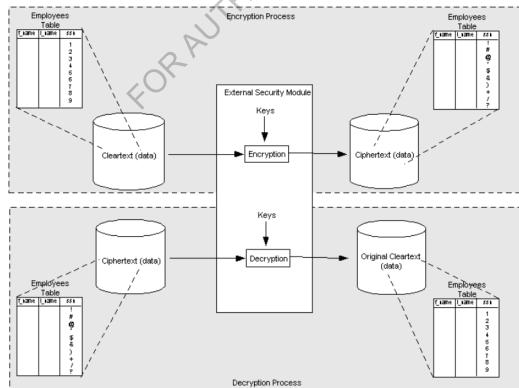
O banco de dados eleito constitui um dos itens de maior relevância neste projeto, não tão somente por ser responsável em armazenar todas as informações do sistema de gerenciamento de conteúdo, como também por ser um dos itens responsáveis pela segurança destes dados. Duas premissas foram imputadas a este item, o *TDE*, (criptografia de dados transparente) que segundo Oracle (2017), permite uma criptografia simples e fácil para dados confidenciais em coluna, sem exigir que o usuário gerencie as chaves de criptografia. Os dados são descriptografados de forma transparente para o usuário, contudo esta segurança somente é garantida com os dados em repouso, durante o transporte os dados precisam ser protegidos de outra forma. Esta proteção será demonstrada na seção dedicada à criptografia.

De forma sucinta pode-se definir o funcionamento da *TDE* da seguinte forma:

Quando uma tabela contém colunas criptografadas, uma única chave é usada para criptografar, independentemente do número de colunas desta tabela, esta chave é criptografada por uma chave mestra e armazenada em uma tabela chamada dicionário no banco de dados. A chave mestra é armazenada fora do banco e somente acessível pelos administradores de segurança do sistema.

Na figura 30 é possível visualizar o diagrama da operação de criptografia *TDE*.

Figura 30 – MySql TDE.



Fonte: Oracle (2017).

A segunda premissa a ser considerada tratando-se do item banco de dados é a necessidade de auditoria do conteúdo armazenado. Neste ponto é preciso considerar que tanto a *TDE* com a auditoria somente estão disponíveis nas versões comerciais dos bancos de dados que foram testados.

Testes foram realizados nos seguintes bancos:

Oracle - Oracle

SQL - Microsoft

Mysql - Oracle versão Comercial

Embora todos os bancos avaliados atendam os requisitos, a escolha foi do banco de dados Mysql. A escolha levou em consideração dos custos envolvidos na aquisição de licença de uso.

De acordo com a Oracle (2017) a função de auditoria é baseada em uma política de fácil uso que permite que a partir do momento em que os dados sensíveis são coletados tornam-se passíveis de auditoria. Isso permite a conformidade com regulamentações internacionais como HIPPA e Sarbanes-Oxley, que não serão abordadas neste dissertação.

Na figura 31 tem-se o diagrama do processo de auditoria usado no MySQL *Enterprise*, onde é possível destacar a habitação do *plugin* de auditoria. Esta dissertação não aprofundará o tema pois não é seu cerne.

Figura 31 – Auditoria.



Fonte: Oracle (2017).

## 4.7 Criptografia Aplicada

Ao longo de toda a dissertação foi explicitado que a segurança do conteúdo seria a um fator preponderante para a realização e aceitação do sistema colaborativo para emissoras de TV a baixo custo. Foram evidenciadas as vantagem da utilização da criptografia de curva elíptica em detrimento a criptografia RSA. A seguir será demonstrado a sua efetiva utilização permeando os vários itens do sistema.

A primeira comprovação da utilização deste algoritmo de criptografia está no acesso às máquinas instanciadas no ambiente em nuvem da AWS, como pode ser visto na figura 32. A autenticação é realizada usando *ECDSA* que é Algoritmo de Assinatura Digital de Curvas Elípticas.

Figura 32 – ECC Autenticação AWS.

```
Jorges-MacBook-Air:~ jorgevarella$ cd Downloads/
Jorges-MacBook-Air:Downloads jorgevarella$ ssh -i "20072017.pem" ubuntu@ec2-18-2
31-44-123.sa-east-1.compute.amazonaws.com
The authenticity of host 'ec2-18-231-44-123.sa-east-1.compute.amazonaws.com (18.
231.44.123)' can't be established.
ECDSA key fingerprint is SHA256:/LUPhwEalW6zXG8Fowia7uy1YoK1R6aZmfAimNF3YBs.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-18-231-44-123.sa-east-1.compute.amazonaws.com,18
.231.44.123' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-1026-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 Get cloud support with Ubuntu Advantage Cloud Guest:
 http://www.ubuntu.com/business/services/cloud

36 packages can be updated.
0 updates are security updates.

Last login: Thu Jul 27 23:17:13 2017 from 189.3.255.178
ubuntu@ip-172-31-30-68:~$
```

Fonte: Amazon (2017).

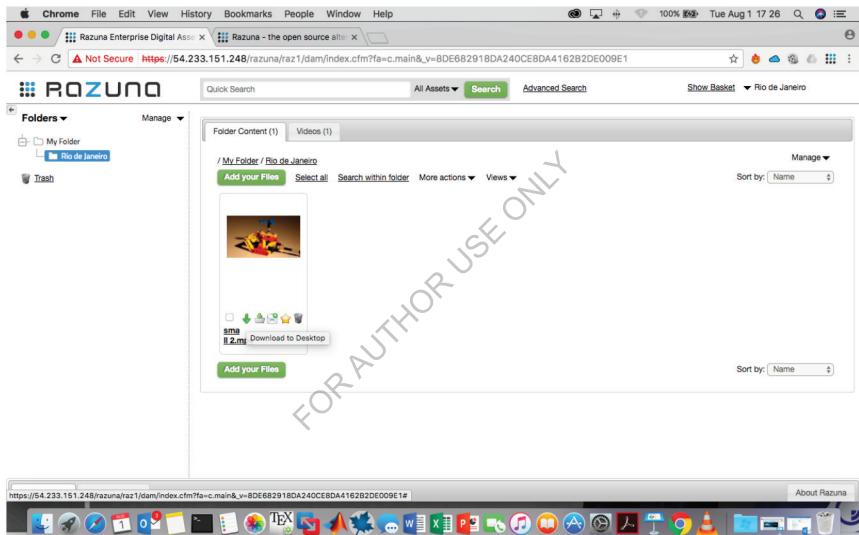
Embora a utilização da autenticação atenda a premissa do uso da criptografia de curva elíptica no sistema, isto encontrava-se no modo nativo de acesso ao ambiente de nuvem da AWS e não representou contribuição de melhoria ao sistema, porém um outro problema encontrado durante a execução do projeto foi a segurança dos dados em trânsito. Isto foi mencionado em capítulo anterior.

Conforme definido, o sistema de controle de conteúdo o *MAM*, da empresa *Razuna*, tem como

característica o uso do ambiente *Web* e faz uso de um servidor *web TomCat*, que segundo a TomCat (2017), é especificamente um servidor *Java*, contudo este servidor não utiliza criptografia na sua comunicação.

A solução para esta deficiência foi a instalação de um *front end*. A escolha foi baseada nas recomendações do desenvolvedor do MAM. As possibilidades seriam o servidor *apache* ou o servidor *Nginx*. Embora os dois fossem similares a escolha foi pelo servidor *Nginx*, devido a simples integração com o ambiente do MAM, como pode ser visto na figura 33.

Figura 33 – ECC Razuna.



Fonte: Razuna (2017).

Na criação deste ambiente com um nível de segurança aprimorado fez-se uso da ferramenta *OpenSSL*, que de acordo com Openssl (2017) é uma biblioteca de criptografia de propósito geral e ideal para os protocolos *Transport Layer Security* (TLS) e *Secure Sockets Layer* (SSL).

A figura 34 mostra a criação do certificado usando criptografia de curva elíptica utilizado

pelo *front end Nginx*. Destaca-se nesta figura a *EDCH* usando a curva P-256, que segundo Gueron e Krasnov (2015), é uma curva recomendada pelo *NIST* e que permite uma perfeita integração com *OpenSSL*.

Esta curva foi mencionada no capítulo Criptografia, no tópico "Escolha da Curva Elíptica", onde encontra-se um aprofundamento mais teórico desta e de outras curvas.

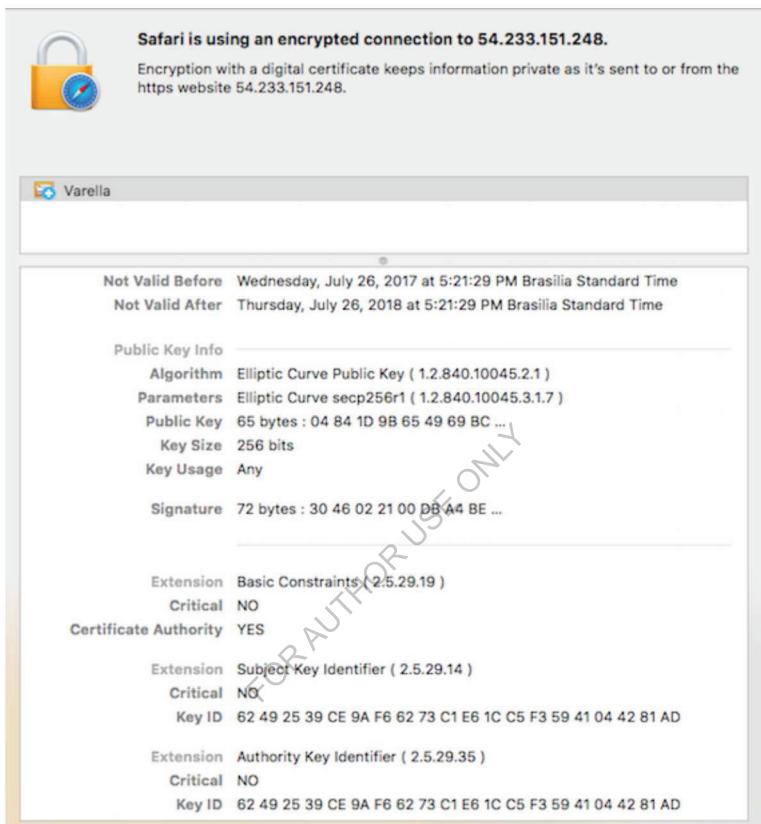
Figura 34 – ECC OpenSSL.

```
No client certificate CA names sent
Peer signing digest: SHA512
Server Temp Key: ECDH, P-256, 256 bits
---
SSL handshake has read 1073 bytes and written 431 bytes
---
New, TLSv1/SSLv3, Cipher is ECDHE-ECDSA-AES256-GCM-SHA384
Server public key is 256 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
SSL-Session:
    Protocol : TLSv1.2
    Cipher   : ECDHE-ECDSA-AES256-GCM-SHA384
    Session-ID: 87C1323011B0EC876CB44916B1C3BCDF939E2D5138AA4A259449192526DE6844
    Session-ID-ctx:
    Master-Key: 77FB09020FBC69399CA80144BB6E5EFC78B3C987746A6A93F78EB3B2ADAB2F8E
2A38A2AB7B6F701BA64E57E0C1F1EDB3
    Key-Ag      : None
    PSK identity: None
    PSK identity hint: None
    SRP username: None
    TLS session ticket lifetime hint: 300 (seconds)
```

Fonte: Print Screen de tela.

A figura 35 tem o objetivo meramente ilustrativo de demonstrar a utilização do certificado construído com a ferramenta *OpenSSL*, onde devem ser destacados o tamanho de chave de 256 bits, a data de criação do certificado e a validade do mesmo.

Figura 35 – ECC OpenSSL Certificado.



Fonte: Print Screen de tela.

## 5 Conclusão

A transição do modelo analógico para o digital no ambiente de *broadcast* e logo após uma nova transição para o modelo de virtualização, causaram grandes mudanças no setor. Novas tecnologias emergiram durante o processo, exigindo dos profissionais aperfeiçoamento para o desempenho de suas atividades.

O ambiente em nuvem, fruto destas novas tecnologia se mostrou extremamente promissor para a descentralização de produções televisivas, contudo a utilização desta nova ferramenta trouxe também problemas relacionados a segurança do conteúdo armazenado, em servidores localizados fora do domínio das emissoras o que provocou certo desconforto e desconfiança no seu uso.

O alto custo de implementação ou adaptação do ambiente em nuvem é um fator que ainda impede seu uso. Sabendo-se que sistemas nesse ambiente são de uso de grandes corporações, a proposta desta dissertação foi desenvolver um ambiente apoiado em uma tecnologia capaz de fornecer soluções a um custo acessível a pequenas emissoras de TV, como por exemplo as TVs comunitárias.

A prototipagem levou em consideração o uso de *software* livre como solução ideal, desde que não compromettesse a segurança dos dados armazenados, para isso fez uso do sistema de controle de mídias Razuna, software que é o coração de todo o sistema. Embora tenha sido necessário fazer uso de uma ferramenta para garantir a segurança dos dados em trânsito, como o *Nginx*, a implementação se mostrou eficiente e durante os testes atendeu perfeitamente os usuários do sistema. Porém, um teste de carga para provocar o estresse do sistema não foi realizado, levando apenas em consideração outros usuários que fazem uso da ferramenta conforme pode ser visto na figura 36.

Clientes do sistema Razuna como a BBC, empresa de TV estatal britânica, a *Tunner* um conglomerado na área de mídias, dentre outros, permitem fazer uma avaliação mesmo que subjetiva da robustez do sistema, embora futuros testes de carga tenham sido planejados para serem realizados.

Figura 36 – Clientes Razuna.

Secure & reliable	Powerful Search Engine	Extensible
 <p>The Razuna DAM will store your digital assets securely. The system is build with security in mind. You can safely store on your network, use SSL encryption, setup user permissions and/or deploy your digital asset management system in the cloud.  <a href="#">Read more...</a></p>	 <p>Every asset within your Digital Asset Management System gets indexed and made available in a powerful search. Furthermore, Razuna DAM supports reading and writing of metadata (XMP, IPTX, EXIF, etc.), be it videos, images, audios or more. <a href="#">Read more...</a></p>	 <p>Razuna, the Open Source Digital Asset Management system, builds on open standards. Use the <a href="#">Wordpress plugin</a> to integrate assets seamlessly to your Wordpress site. <a href="#">Razuna Desktop</a> further supports the ease of use. Lastly, <a href="#">developers</a> can use the API to expand further.</p>



Fonte: Razuna (2017).

A segurança do sistema, conforme mencionado diversas vezes durante a dissertação, foi fator norteador dos passos quanto à escolha do banco de dados e da criptografia a serem implementados. No caso específico do banco, fez-se uso do *mysql*, porém vale mais uma vez ressaltar que a versão utilizada foi a versão comercial, cuja licença anual é de aproximadamente 5.000 dólares, somente esta versão contém as funcionalidades de *TDE* e auditoria.

No caso da criptografia, conforme explicitado no capítulo dedicado ao tema, a utilização da criptografia de curva elíptica se deve ao fato do menor consumo de energia, comparado à criptografia RSA e a utilização de dispositivos móveis para *upload* de material. A curva utilizada foi a P-256 indicada pelo NIST e segundo Gryb (2017) é contestada no meio acadêmico devido a sua fragilidade. Há outras curvas que poderão ser implementadas no sistema sem restrições.

Os custos deste projeto seriam atribuídos ao serviços de nuvem, o licenciamento do banco de dados e ao *playout* a ser definido por cada emissora participante, o valor é estimado em torno de 600 dólares mês, considerando o uso do *software Just Play* usado nos testes do sistema. Estes valores são significativamente menores comparados aos custos envolvidos somente com a produção tradicional e

que pode alcançar cifras vultuosas.

Com essas considerações conclui-se que é possível construir um sistema em nuvem de baixo custo destinado a emissoras de TV com pouco capital para investimento na área de produção e que desejam compartilhar conteúdo, formando um grupo capaz de dividir material entre seus membros e com isso ter uma programação sempre atualizada diminuindo assim a quantidade de reprises de programas, fato que afasta o telespectador do canal em questão.

## 5.1 Trabalhos Futuros

O sistema do MAM proposto não possui versão em português, com isso vale lembrar que a licença de uso incentiva a melhoria continua do produto e a sua tradução traria benefícios significativos aos usuários do sistema.

Apesar da criptografia de curva elíptica se apresentar como uma solução eficiente para o problema proposto, deve-se considerar que a segurança dos dados em repouso e em transito usam dois tipos de criptografia. Um algoritmo de criptografia único que possa garantir os dados em transito e em repouso é uma melhoria que facilitaria a implementação e reduziria o custo com a compra de versões comercias do banco de dados. Entendesse nesta linha que a criptografia pós quântica, holomórfica poderá resolver esta questão.

Da mesma forma quanto a melhoria da segurança é possível vislumbrar a utilização dos item em nuvem hibrida, onde os dados estariam na nuvem privada e os vídeos em uma ambiente de nuvem pública, esta solução estaria alinhada com as soluções comercias quanto ao nível de segurança do sistema.

## Referências

- ALECRIM, E. Assinatura digital e certificação digital. São Paulo, 2005. Disponível em: <<https://www.infowester.com/index.php>>. Acesso em: 21 abr 2017. Citado na página 29.
- AMAZON. *Tipos de computação em nuvem*. USA, 2017. Disponível em: <<https://aws.amazon.com/pt/types-of-cloud-computing/>>. Acesso em: 21 abr 2017. Citado 5 vezes nas páginas 25, 53, 54, 56 e 75.
- BARBOSA, J. C. Criptografia de chave pública baseada em curvas elípticas. *Monografia final de curso de mestrado em redes*. Rio de Janeiro: COPPE/UFRJ, 2003. Citado 2 vezes nas páginas 34 e 40.
- BLAKE SEROUSSI, S. Elliptic in cryptography. Cambridge, Londres., 1999. Citado na página 39.
- BROWN, M. et al. Software implementation of the nist elliptic curves over prime fields. In: SPRINGER. *Cryptographers' Track at the RSA Conference*. [S.I.], 2001. p. 250–265. Citado na página 49.
- CORREIA, S. *Criptografia via curvas elípticas*. Rio de Janeiro, 2011. Disponível em: <<http://www2.unirio.br/unirio/ccet/profmat/tcc/2011/TCCSergioCorreia.pdf>>. Acesso em: 01 jun 2017. Citado 3 vezes nas páginas 43, 44 e 45.
- CUNHA, M. F. da. *Virtualização de Estúdios Móveis na Produção de conteúdo Audiovisuais*. [S.I.]: Faculdade de Engenharia do Porto, 2016. Citado na página 20.
- DAHAB, R.; LÓPEZ-HERNÁNDEZ, J. C. Técnicas criptográficas modernas: algoritmos e protocolos. *Instituto de Computação-UNICAMP*, p. 30–31, 2007. Citado na página 41.
- DUSTED.CODES. *The beauty of asymmetric encryption - RSA crash course for developers*. USA, 2015. Disponível em: <<https://dusted.codes/the-beauty-of-asymmetric-encryption-rsa-crash-course-for-developers>>. Acesso em: 21 abr 2017. Citado 2 vezes nas páginas 30 e 33.
- FFMPEG. *FFMPEG*. USA, 2017. Disponível em: <<https://www.ffmpeg.org>>. Acesso em: 20 ago 2017. Citado na página 63.
- GARFINKEL, S.; SPAFFORD, G. Practical unix & internet security. O'Reilly & Associates, Inc., 1999. Citado na página 33.
- GNU. *AGPL*. USA, 2017. Disponível em: <<https://www.gnu.org/licenses/agpl-3.0.html>>. Acesso em: 20 ago 2017. Citado na página 58.
- GOLDWASSER, S. K. Primality testing using elliptic curves. ACM, v. 46, n. 4., 1999. Citado na página 40.

- GRYB, O. *Criptografia de Curva Elíptica*. USA, 2017. Disponível em: <<http://ogryb.blogspot.com.br/2014/11/why-i-dont-trust-nist-p-256.html>>. Acesso em: 21 abr 2017. Citado na página 80.
- GUERON, S.; KRASNOV, V. Fast prime field elliptic-curve cryptography with 256-bit primes. *Journal of Cryptographic Engineering*, Springer, v. 5, n. 2, p. 141–151, 2015. Citado na página 76.
- GUPTA, V. et al. Performance analysis of elliptic curve cryptography for ssl. In: ACM. *Proceedings of the 1st ACM workshop on Wireless security*. [S.I.], 2002. p. 87–94. Citado 3 vezes nas páginas 46, 48 e 49.
- HAZAY, C. et al. Concurrently-secure blind signatures without random oracles or setup assumptions. In: SPRINGER. *Theory of Cryptography Conference*. [S.I.], 2007. p. 323–341. Citado na página 48.
- HWANG, K.; DONGARRA, J.; FOX, G. C. *Distributed and cloud computing: from parallel processing to the internet of things*. [S.I.]: Morgan Kaufmann, 2013. Citado 3 vezes nas páginas 23, 24 e 25.
- IBM. *Modelos de serviços de Computação em nuvem*. São Paulo, 2017. Disponível em: <<https://www.ibm.com/developerworks/br/cloud/library/cl-cloudservices1iaas/>>. Acesso em: 28 abr 2017. Citado na página 26.
- JAIN, N.; JAIN, P.; KAPIL, N. Enhanced data security model for cloud using ecc algorithm and third party auditor. Citado na página 57.
- JUELS, A.; JR, B. S. K. Pors: Proofs of retrievability for large files. In: ACM. *Proceedings of the 14th ACM conference on Computer and communications security*. [S.I.], 2007. p. 584–597. Citado na página 27.
- KLIMA, R.; SIGMON, N. P.; STITZINGER, E. *Applications of abstract algebra with Maple and MATLAB*. [S.I.]: CRC Press, 2006. Citado na página 47.
- KOBLITZ, N. Elliptic curve cryptosystems. *Mathematics of Computation*, v. 43, n. 177, p., 1987. Citado 2 vezes nas páginas 39 e 50.
- KRUTZ, R. L.; VINES, R. D. *Cloud security: A comprehensive guide to secure cloud computing*. [S.I.]: Wiley Publishing, 2010. Citado 3 vezes nas páginas 25, 26 e 27.
- LADEIRA, L. Z. *Canais laterais em criptografia simétrica e de curva elíptica: ataques de contramedidas*. Rio de Janeiro, 2016. Disponível em: <<http://sbseg2016.ic.uff.br/pt/index.php>>. Acesso em: 21 maio 2017. Citado 2 vezes nas páginas 49 e 50.
- LENSTRA, H. A. Factoring integers with elliptic curves. *N Math.(2)*, v. 126, p. 649–67., 1987. Citado na página 40.
- LOIDREAU, P.; SENDRIER, N. Weak keys in the mceliece public-key cryptosystem. *IEEE Transactions on Information Theory*, IEEE, v. 47, n. 3, p. 1207–1211, 2001. Citado na página 29.

- MILLER, V. Use of elliptic curves in cryptography. *Advances in Criptology - CRYPTO 85*, v. 468, p. 417–426., 1986. Citado na página 39.
- MORENO PEREIRA, D. Criptografia em software e hardware. Novatec., 2005. Citado 2 vezes nas páginas 31 e 34.
- OPENSSL. *OpenSSL*. USA, 2017. Disponível em: <<https://www.openssl.org/>>. Acesso em: 20 ago 2017. Citado na página 76.
- ORACLE. *TDE*. USA, 2017. Disponível em: <[https://docs.oracle.com/cd/B19306\\_01/network.102/b14268/asotrans](https://docs.oracle.com/cd/B19306_01/network.102/b14268/asotrans)>. Acesso em: 22 ago 2017. Citado 2 vezes nas páginas 73 e 74.
- POPEK, G. J.; GOLDBERG, R. P. Formal requirements for virtualizable third generation architectures. *Communications of the ACM*, ACM, v. 17, n. 7, p. 412–421, 1974. Citado na página 22.
- RAZUNA. *MAM*. USA, 2017. Disponível em: <<http://razuna.org>>. Acesso em: 20 ago 2017. Citado 12 vezes nas páginas 58, 60, 61, 62, 63, 64, 65, 67, 68, 69, 76 e 80.
- RIVEST, R. L.; SHAMIR, A.; ADLEMAN, L. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, ACM, v. 21, n. 2, p. 120–126, 1978. Citado 2 vezes nas páginas 34 e 35.
- SILVA, B. R. *Uma Técnica de Análise Estática para Detecção de Canais Laterais Baseados em Tempo*. Minas Gerais, 2015. Disponível em: <[http://homepages.dcc.ufmg.br/~fernando/publications/papers\\_pt/RodriguesSilva15SSB.pdf](http://homepages.dcc.ufmg.br/~fernando/publications/papers_pt/RodriguesSilva15SSB.pdf)>. Acesso em: 21 maio 2017. Citado na página 50.
- SILVA, T. H.; MACHARET, D. G.; TEIXEIRA, C. F. Análise do desempenho de algoritmos criptográficos em dispositivos móveis. In: *Wperformance Workshop de Desempenho de Sistemas Computacionais e de Comunicação, Anais do XXVIII Congresso da SBC, Belém*. [S.I.: s.n.], 2008. Citado na página 29.
- TANENBAUM, A. S. *Redes de computadoras*. [S.I.]: Rio de Janeiro: Editora Campus, 2003. Citado na página 32.
- TKOTZ, V. Criptografia: segredos embalados para viagem. *São Paulo: Novatec*, p. 16, 2005. Citado na página 37.
- TOMCAT. *TomCat*. USA, 2017. Disponível em: <<https://tomcat.apache.org>>. Acesso em: 31 ago 2017. Citado na página 76.
- TOOLONAIR. *JustPlay*. USA, 2017. Disponível em: <<http://www.toolsonair.com>>. Acesso em: 20 ago 2017. Citado 2 vezes nas páginas 71 e 72.
- UFGRS, L. *Resiliência*. Rio Grande do Sul, 2017. Disponível em: <[https://www.lume.ufrgs.br/bitstream/handle/10183/112327/Poster\\_36018.pdf?sequence=2](https://www.lume.ufrgs.br/bitstream/handle/10183/112327/Poster_36018.pdf?sequence=2)>. Acesso em: 29 ago 2017. Citado na página 67.

- WANG, C. et al. Privacy-preserving public auditing for data storage security in cloud computing. In: IEEE. *Infocom, 2010 proceedings ieee*. [S.I.], 2010. p. 1–9. Citado na página 27.
- WANG, Q. et al. Enabling public verifiability and data dynamics for storage security in cloud computing. In: SPRINGER. *European symposium on research in computer security*. [S.I.], 2009. p. 355–370. Citado na página 27.
- XING, Y.; ZHAN, Y. Virtualization and cloud computing. In: *Future Wireless Networks and Information Systems*. [S.I.]: Springer, 2012. p. 305–312. Citado na página 22.
- YOKOYAMA, D. M. *Modelo para o Escalonamento de Aplicacoes Cientificas em Ambiente de Nuvens Baseado em Afinidade*. [S.I.]: Dissertacao de Mestrado LNCC, 2015. Citado na página 22.
- ZHANG, Q.; CHENG, L.; BOUTABA, R. Cloud computing: state-of-the-art and research challenges. *Journal of internet services and applications*, Springer, v. 1, n. 1, p. 7–18, 2010. Citado na página 22.

FOR AUTHOR USE ONLY

FOR AUTHOR USE ONLY

# APÊNDICE A – Programa em Maple - Criptografia RSA.

*restart*

#Mensagem a ser transmitida#

```
mensagem[1] := '%Esternocleidomastoideo%';
%Esternocleidomastoideo%
```

(1)

```
conv := convert(mensagem[1], bytes) :
conv,
[37, 69, 115, 116, 101, 114, 110, 111, 99, 108, 101, 105, 100, 111, 109, 97, 115, 116, 111, 105,
100, 101, 111, 37]
```

(2)

*n := 437 :*

*e := 13 :*

*d := 61 :*

#  $c = m^e \pmod{n}$ ; Criptografando#

```
cripto := map(x → x^e mod n, conv)
[379, 69, 115, 231, 403, 114, 48, 214, 66, 72, 403, 146, 340, 214, 401, 136, 115, 231, 214, 146,
340, 403, 214, 379]
```

(3)

#Verificação do texto criptografado#

```
f := convert(cripto, bytes);
***
```

(4)

#  $md = m^d \pmod{n}$ ; Decriptografando#

```
Decripto := map(y → y^d mod n, cripto)
[37, 69, 115, 116, 101, 114, 110, 111, 99, 108, 101, 105, 100, 111, 109, 97, 115, 116, 111, 105,
100, 101, 111, 37]
```

(5)

[> #Verificação do texto decriptografado#
 $g := convert(\%, bytes);$

"%Esternocleidomastoideo%"

(6)

FOR AUTHOR USE ONLY

# APÊNDICE B – Programa em Maple - Implementação ECC - Elgamal4.mpl

```

> # Impletacao de criptografia de curvas elipticas
> #
> # variaveis globais elip_c elip_d
> elip_c := c;
> elip_d := d;
> elip_p := p;
> #k:=4;

> SOMA := proc(P::list,Q::list)
> local x1,x2,x3,y1,y2,y3;
> if nargs>2 then
>   return SOMA(SOMA(P,Q),args[3..nargs]);
> end if;
> if member(0,{P,Q}) then return op({P,Q} minus {0}) fi;
> x1,y1:=op(P);
> x2,y2:=op(Q);
> if x1=x2 and y1+y2 mod elip_p=0 then return 0 fi;
> if P=Q then
>   x3 := ((3*x1**2+ elip_c)/(2*y1))^2-x1-x2 mod elip_p;
>   y3 := ((3*x1**2+ elip_c)/(2*y1))*(x1-x3)-y1 mod elip_p;
> else
>   x3 := ((y2-y1)/(x2-x1)) ^2-x1-x2 mod elip_p;
>   y3 := ((y2-y1)/(x2-x1))*(x1-x3) -y1 mod elip_p;
> fi;
> #
> return [x3,y3];
> end proc;

> MUL := proc(k,P)
>   if k=1 then P else SOMA(P$ k) fi
> end proc;

> DIV := proc(P,Q)
> local i;
> for i to 10000 do if MUL(i,Q)=P then return i fi od;
> if i>10000 then error "exceeded limit" else i fi;
> end;

> ORDEM := proc(P)
> local k;
> for k do
>   if not type(traperror(MUL(k,P)),list) then return k fi;
> od;
> end;

> MENOS := proc(P::list)
> if P=0 then P else [op(1,P),-op(2,P)] fi;
> end;

```

```
> CONV := proc(texto,n)
> local L;
> if nargs=1 then
>   convert(texto,'bytes')
> elif nargs=2 then
>   L:=procname(texto);
>
> else
>   error "expecting two arguments, got %",nargs
> end if;
> end proc:
```

FOR AUTHOR USE ONLY

FOR AUTHOR USE ONLY

# APÊNDICE C – Programa em Maple -

## Método de Criptografia ECC

```

MÉTODO DE CRIPTOGRAFIA DE CURVAS ELÍTICAS
> restart
> read "Elgamal4.mpl"

> c, d, p := 1, 6, 19;  4·c³ + 27·d² mod p
c, d, p := 1, 6, 19
7                                         (1)

Receptor da mensagem
> a := [0, 5]                                a := [0, 5]                         (2)
> n := 4                                     n := 4                               (3)
> b := MUL(n, a)                            b := [3, 6]                         (4)
Envio da mensagem (usando os valores a e b enviados pelo receptor da
mensagem)
> w := [18, 17] # esta é a mensagem
w := [18, 17]                                 (5)
> k := 3                                     k := 3                               (6)
> y := MUL(k, a)                            y := [2, 4]                         (7)
> z := SOMA(w, MUL(k, b))                  z := [14, 3]                         (8)
w e z são enviados. Para decifrar
> w, z
[18, 17], [14, 3]                           (9)
> SOMA(z, MENOS(MUL(n, y)))
[18, 17]                                       (10)

GENERALIZAÇÃO - Texto de tamanho arbitrário
> texto :=
  "isto é um teste do método de criptografia de curvas
  elíticas"
  texto := "isto é um teste do método de criptografia de curvas elíticas" (11)
> texto_numerico := CONV(texto)
texto_numerico := [105, 115, 116, 111, 32, 195, 169, 32, 117, 109, 32, 116, 101, 115, 116,
101, 32, 100, 111, 32, 109, 195, 169, 116, 111, 100, 111, 32, 100, 101, 32, 99, 114, 105,
112, 116, 111, 103, 114, 97, 102, 105, 97, 32, 100, 101, 32, 99, 117, 114, 118, 97, 115,
32, 101, 108, 195, 173, 116, 105, 99, 97, 115]
> c, d, p := 1, 6, 503;  4·c³ + 27·d² mod p
c, d, p := 1, 6, 503
473                                         (13)

Receptor da mensagem
> a := [7, 1]                                a := [7, 1]                         (14)
> n := 4

```

```

n := 4                                (15)
> b := MUL(n, a)                      b := [432, 265]                         (16)
Envio da mensagem (usando os valores a e b enviados pelo receptor da
mensagem)
> P := [1, 6]                          P := [1, 6]                               (17)
> texto_ECC := map(k->MUL(k, P), texto_numerico)
texto_ECC := [[473, 151], [275, 247], [377, 203], [162, 353], [390, 344], [395, 333],
[154, 490], [390, 344], [356, 109], [65, 452], [390, 344], [377, 203], [285, 387],
[275, 247], [377, 203], [285, 387], [390, 344], [160, 312], [162, 353], [390, 344],
[65, 452], [395, 333], [154, 490], [377, 203], [162, 353], [160, 312], [162, 353],
[390, 344], [160, 312], [285, 387], [390, 344], [346, 322], [189, 27], [473, 151],
[123, 459], [377, 203], [162, 353], [192, 449], [189, 27], [169, 336], [313, 461],
[473, 151], [169, 336], [390, 344], [160, 312], [285, 387], [390, 344], [346, 322],
[356, 109], [189, 27], [362, 337], [169, 336], [275, 247], [390, 344], [285, 387],
[25, 463], [395, 333], [283, 18], [377, 203], [473, 151], [346, 322], [169, 336],
[275, 247]]                                (18)
> k := 4                                  k := 4                                (19)
> y := MUL(k, a)                      y := [432, 265]                         (20)
> texto_cifrado := map(w->SOMA(w, MUL(k, b)), texto_ECC)
texto_cifrado := [[469, 255], [461, 4], [315, 251], [162, 150], [27, 87], [31, 72], [421,
249], [27, 87], [225, 54], [393, 103], [27, 87], [315, 251], [249, 4], [461, 4], [315,
251], [249, 4], [27, 87], [325, 373], [162, 150], [27, 87], [393, 103], [31, 72], [421,
249], [315, 251], [162, 150], [325, 373], [162, 150], [27, 87], [325, 373], [249, 4],
[27, 87], [429, 43], [163, 251], [469, 255], [452, 218], [315, 251], [162, 150], [135,
463], [163, 251], [368, 324], [74, 33], [469, 255], [368, 324], [27, 87], [325, 373], [249,
4], [27, 87], [429, 43], [225, 54], [163, 251], [87, 458], [368, 324], [461, 4], [27,
87], [249, 4], [288, 32], [31, 72], [89, 415], [315, 251], [469, 255], [429, 43], [368,
324], [461, 4]], [14, 3]                  (21)
w e z são enviados. Para decifrar
> texto_cifrado, z
[[469, 255], [461, 4], [315, 251], [162, 150], [27, 87], [31, 72], [421, 249], [27, 87],
[225, 54], [393, 103], [27, 87], [315, 251], [249, 4], [461, 4], [315, 251], [249, 4],
[27, 87], [325, 373], [162, 150], [27, 87], [393, 103], [31, 72], [421, 249], [315,
251], [162, 150], [325, 373], [162, 150], [27, 87], [325, 373], [249, 4], [27, 87],
[429, 43], [163, 251], [469, 255], [452, 218], [315, 251], [162, 150], [135, 463],
[163, 251], [368, 324], [74, 33], [469, 255], [368, 324], [27, 87], [325, 373], [249,
4], [27, 87], [429, 43], [225, 54], [163, 251], [87, 458], [368, 324], [461, 4], [27,
87], [249, 4], [288, 32], [31, 72], [89, 415], [315, 251], [469, 255], [429, 43], [368,
324], [461, 4]], [14, 3]                (22)

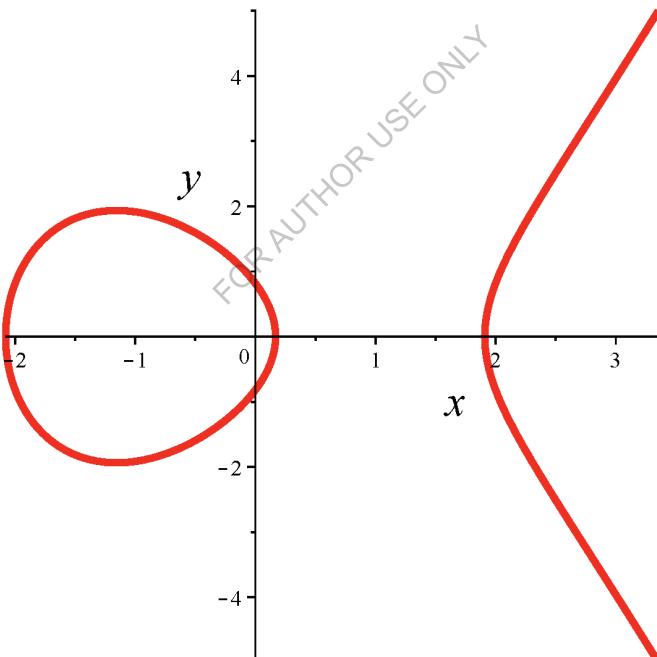
```

```
> texto_decifrado := map(z→SOMA(z, MENOS(MUL(n, y))), texto_cifrado) (23)
texto_decifrado := [[473, 151], [275, 247], [377, 203], [162, 353], [390, 344], [395,
333], [154, 490], [390, 344], [356, 109], [65, 452], [390, 344], [377, 203], [285,
387], [275, 247], [377, 203], [285, 387], [390, 344], [160, 312], [162, 353], [390,
344], [65, 452], [395, 333], [154, 490], [377, 203], [162, 353], [160, 312], [162,
353], [390, 344], [160, 312], [285, 387], [390, 344], [346, 322], [189, 27], [473,
151], [123, 459], [377, 203], [162, 353], [192, 449], [189, 27], [169, 336], [313,
461], [473, 151], [169, 336], [390, 344], [160, 312], [285, 387], [390, 344], [346,
322], [356, 109], [189, 27], [362, 337], [169, 336], [275, 247], [390, 344], [285,
387], [25, 463], [395, 333], [283, 18], [377, 203], [473, 151], [346, 322], [169,
336], [275, 247]] (24)
> texto_dec_num := map(DIV, texto_decifrado, P)
texto_dec_num := [105, 115, 116, 111, 32, 195, 169, 32, 117, 109, 32, 116, 101, 115, 116,
101, 32, 100, 111, 32, 109, 195, 169, 116, 111, 100, 111, 32, 100, 101, 32, 99, 114, 105,
112, 116, 111, 103, 114, 97, 102, 105, 97, 32, 100, 101, 32, 99, 117, 114, 118, 97, 115,
32, 101, 108, 195, 173, 116, 105, 99, 97, 115] (25)
> texto_original := convert(texto_dec_num, bytes)
    texto_original := "isto é um teste do método de criptografia de curvas elíticas"
```

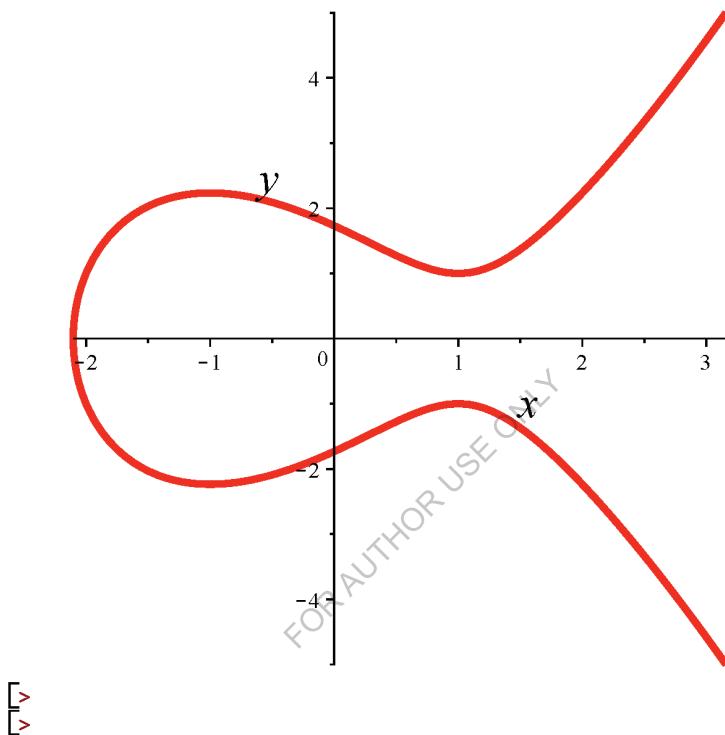
FOR AUTHOR USE ONLY

# APÊNDICE D – Programa em Maple - Curvas Elípticas.

```
> eq1 :=  $y^2 = x^3 - 4 \cdot x + 0.67$ 
      
$$eq1 := y^2 = x^3 - 4x + 0.67$$
 (1)
> plots[implicitplot](eq1, x = -7..10, y = -5..5, numpoints = 10000, axesfont = [10, 10],
    labelfont = [20, 20], thickness = 4, color = red)

>
> eq2 :=  $y^2 = x^3 - 3 \cdot x + 3$ 
      
$$eq2 := y^2 = x^3 - 3x + 3$$
 (2)
plots[implicitplot](eq2, x = -7..10, y = -5..5, numpoints = 10000, axesfont = [10, 10], labelfont
= [20, 20], thickness = 4, color = red)
```









# yes I want morebooks!

Buy your books fast and straightforward online - at one of world's fastest growing online book stores! Environmentally sound due to Print-on-Demand technologies.

Buy your books online at  
**[www.morebooks.shop](http://www.morebooks.shop)**

Compre os seus livros mais rápido e diretamente na internet, em uma das livrarias on-line com o maior crescimento no mundo!  
Produção que protege o meio ambiente através das tecnologias de impressão sob demanda.

Compre os seus livros on-line em  
**[www.morebooks.shop](http://www.morebooks.shop)**

KS OmniScriptum Publishing  
Brivibas gatve 197  
LV-1039 Riga, Latvia  
Telefax: +371 686 20455

[info@omnascriptum.com](mailto:info@omnascriptum.com)  
[www.omnascriptum.com](http://www.omnascriptum.com)

OMNI**S**criptum







