

Wellington Reina Barbieri

Proposta e modelagem de comunicação segura para redes veiculares (VANETs)

Dissertação apresentada ao Programa de Pós-Graduação de Ciências de Computação do Instituto de Biociências, Letras e Ciências Exatas, da Universidade Estadual Paulista “Júlio de Mesquita Filho”, como parte dos requisitos necessários para obtenção do título de Mestre em Ciência da Computação.

São José do Rio Preto
2012

Wellington Reina Barbieri

Proposta e modelagem de comunicação segura para redes
veiculares (VANETs)

Dissertação apresentada ao Programa de Pós-Graduação de Ciências de Computação do Instituto de Biociências, Letras e Ciências Exatas, da Universidade Estadual Paulista “Júlio de Mesquita Filho”, como parte dos requisitos necessários para obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Adj. Marcos Antônio Cavenaghi

Co-Orientador: Prof^a. Dr^a. Roberta Spolon

São José do Rio Preto
2012

Wellington Reina Barbieri

Proposta e modelagem de comunicação segura para redes veiculares (VANETs)

Dissertação apresentada ao Programa de Pós-Graduação de Ciências de Computação do Instituto de Biociências, Letras e Ciências Exatas, da Universidade Estadual Paulista “Júlio de Mesquita Filho”, como parte dos requisitos necessários para obtenção do título de Mestre em Ciência da Computação.

BANCA EXAMINADORA

Prof. Adj. Marcos Antônio Cavenaghi
UNESP – Bauru - Orientador

Prof^o Adj. Aparecido Nilceu Marana
UNESP – Bauru

Prof^a. Dr^a. Sarita Mazzini Bruschi
USP - São Carlos

São José do Rio Preto, 30 de Novembro de 2012

AGRADECIMENTOS

Ao Prof. Dr. Marcos Antonio Cavenaghi que abriu as portas do conhecimento e do mestrado para mim e à Prof^a. Dr^a Roberta Spolon pelo apoio.

Aos meus queridos pais Helio Barbieri e Maria Reina Barbieri que sempre estiveram ao meu lado em todos os momentos difíceis e nas dúvidas, aos meus irmãos que sempre me apoiaram a continuar buscando meus objetivos.

À Prof^a. Dr^a Cristiane Alexandra Lázaro pelo suporte com os cálculos e pelo apoio, não deixando em momento algum desanimar, me escutando e abrindo meus olhos para contratempos e caminhos errados que por algum motivo surgiram no desenvolvimento do mestrado.

Ao Eduardo Garcia por sempre que necessário me substituir nos horários de trabalho quando precisei de tempo para estudar e ou participar de reuniões com meus orientadores.

À Renata Mussio Quintanilha, minha namorada, que sempre esteve ao meu lado me ajudando e apoiando. Pela sua compreensão, pois por diversas vezes não foi possível estar ao seu lado por estar estudando.

Agradeço o apoio de diversas pessoas que, direta ou indiretamente me ajudaram, entre elas estão os funcionários da Biblioteca, do Departamento de Matemática e do Departamento de Ciência da Computação da Unesp/Bauru, a equipe de desenvolvimento e suporte da AD Corretora de Seguros e alguns familiares, entre outros.

*“A mente que se abre a uma nova ideia jamais
voltará ao seu tamanho original.”
(Albert Einstein)*

Resumo

Os avanços constantes da tecnologia, a facilidade da troca de informação em todas as áreas conhecidas e equipamentos cada vez mais sofisticados apresentam um cenário onde se torna imprescindível uma infraestrutura que proporcione comunicação entre os veículos que circulam nas cidades e rodovias, para haver uma maior segurança no tráfego. Esta ferramenta conhecida como VANET (*Vehicular Ad Hoc NETWORK*) é uma subclasse da redes *Ad Hoc* que tem como objetivo a comunicação entre veículos e/ou entre veículos e uma infraestrutura de acostamento, e tem a finalidade de proporcionar segurança e autonomia nas vias rodoviárias. Mas esta troca de informação ainda é frágil e cada vez mais fácil de ser interceptada. Com o intuito de minimizar as invasões e interceptações, foi proposta a utilização do protocolo TESLA (*Timed Efficient Stream Loss-Tolerant Authentication*) que é um protocolo de segurança utilizado na troca de informações nas comunicações e baseia-se na perda de tempo de sincronização entre o emissor e o receptor. O objetivo deste trabalho é modelar em UML e Fluxograma a alteração na comunicação na rede VANET, sendo possível escolher qual protocolo será utilizado e a segurança no protocolo TESLA, que tornará possível a escolha de qual criptografia utilizar, e para esta modelagem está prevista a utilização da técnica da Criptografia das Curvas Elípticas – ECC.

Abstract

The constant advances in technology, the ease of information exchange in all areas, and increasingly sophisticated equipment present a scenario where it becomes essential to have an infrastructure that provides communication between vehicles traveling in cities and highways to be safer in traffic. This tool is known as VANET (Vehicular Ad hoc NETwork) and is a subclass of ad hoc networks that aims communication between vehicles and/or between vehicles and roadside infrastructure, in order to provide security and autonomy roads. But, this exchange of information is still fragile and increasingly easy to be intercepted. In order to minimize intrusions and hijacks, it was proposed to use the TESLA protocol (Timed Efficient Stream Loss-Tolerant Authentication), which is a security protocol used in the communication exchange and based on the loss of synchronization time between transmitter and receiver. The objective of this work is to model in UML and Flowchart change in the VANET network communication, and can choose which protocol will be used and safety at TESLA protocol, which make it possible to choose which encryption to use, and this model is expected to use the technique of Elliptic Curve Cryptography - ECC.

Sumário

Resumo	vi
Abstract.....	vii
Lista de Figuras.....	3
Lista de Tabelas	4
Lista de Abreviaturas e Siglas.....	5
1. Introdução	6
1.1 Motivação e Justificativas.....	8
1.2 Objetivo.....	8
1.3 Organização do texto	9
2. Redes sem fio <i>Ad Hoc</i>	10
2.1 Considerações iniciais.....	10
2.2 Redes Ad Hoc	10
2.3 Vantagens e Desvantagens.....	11
2.4 Redes veiculares Ad Hoc – VANET.....	12
2.4.1 Características de uma rede veicular.....	13
2.4.2 Modelo de uma VANET.....	15
2.4.2.1 Mensagens de Segurança	16
2.4.3 Protocolo para Mensagens de Segurança.....	17
2.4.4 Ataques em Redes Veiculares.....	18
2.4.4.1 Modelo de Atacante	18
2.4.4.2 Possíveis Ataques.....	18
2.4.4.3 Requisitos de Segurança em Redes VANET	19
2.4.5 Ferramentas para Simular VANET.....	20
2.4.6 Segurança em Redes Veiculares	22
2.4.6.1 Relações de Confiança	22
2.4.6.2 TESLA	23
2.4.6.3 TESLA++.....	24

2.5	Considerações Finais	26
3.	Ferramentas para Segurança	27
3.1	Considerações iniciais.....	27
3.2	Criptografia.....	27
3.2.1	Criptografia de Chave Assimétrica.....	29
3.2.2	Criptografia de Chave Simétrica.....	29
3.3	Criptografia Com Curvas Elípticas (ECC).....	29
3.3.1	Método proposto para o desenvolvimento da Curva Elíptica	31
3.3.2	Desenvolvimento do Cálculo da Curva Elíptica	33
3.4	Logaritmo Discreto	35
3.4.1	Método Pollard.....	35
3.5	Considerações Finais	39
4.	Proposta de um protocolo para comunicação em redes veiculares	41
4.1	Considerações iniciais.....	41
4.2	Representação Simplificada.....	41
4.3	Modelagem em UML.....	44
4.3.1	Diagrama de Classes	45
4.3.2	Diagrama de Caso de Uso.....	49
4.3.3	Diagrama de Sequência.....	52
4.3.4	Diagrama de Estado	53
4.4	Fluxograma	68
4.5	Estrutura e desenvolvimento.....	75
4.6	Considerações Finais	76
5.	Conclusão e Trabalhos Futuros.....	77
5.1	Conclusão.....	77
5.2	Trabalhos Futuros	78
6.	Referências Bibliografia	80
	APÊNDICE A – Código do desenvolvimento das ECC.....	86
	APÊNDICE B – Diagrama de Estado Completo da Melhoria.....	91

Lista de Figuras

Figura 1 - Comunicação de redes <i>Ad Hoc</i> veiculares (A) Troca direta de informações entre veículos. (B) Troca de informações entre veículos por meio de infraestrutura.....	13
Figura 2 - Representação do envio da informação e da Mac para o receptor no TESLA.....	24
Figura 3 - Representação do envio do MAC curto e em seguida a informação para o receptor no TESLA++.....	25
Figura 4 - Segurança X Tamanho de Chave (ECC, RSA e DSA)(BARBOSA, 2003).....	31
Figura 5 - Representação da rede VANET	42
Figura 6 - Ilustração da estrutura da VANET.....	43
Figura 7 - Demonstração da alteração da comunicação entre os Nós e Pista na rede VANET.....	44
Figura 8 - Representação das classes utilizadas nas melhorias.....	46
Figura 9 - Representação das classes utilizadas nas melhorias.....	47
Figura 10 - Representação da Alteração do comportamento da Classe VANET.....	48
Figura 11 - Representação das funções recebe mensagem.....	50
Figura 12 - Representação das funções envia mensagem.....	51
Figura 13 - Representação da sequência dos objetos.....	52
Figura 14 - Representação do diagrama de estado - Parte 1	53
Figura 15 - Representação do diagrama de estado - Parte 2	55
Figura 16 - Representação do diagrama de estado - Parte 3	57
Figura 17 - Representação do diagrama de estado - Parte 4	59
Figura 18 - Representação do diagrama de estado - Parte 5	60
Figura 19 - Representação do diagrama de estado - Parte 6	62
Figura 20 - Representação do diagrama de estado - Parte 7	64
Figura 21 - Representação do diagrama de estado - Parte 8	66
Figura 22 - Representação do diagrama de estado - Parte 9	67
Figura 23 - Representação do diagrama de estado - Parte 10	68
Figura 24 - Fluxograma das redes VANETs.....	70
Figura 25 - Fluxograma TESLA Envia Mensagem	71
Figura 26 - Fluxograma TESLA Recebe Mensagem.....	72
Figura 27 - Fluxograma das Mensagens Recebidas no Nó.....	73
Figura 28 - Fluxograma das Mensagens Recebidas na Pista	74
Figura 29 - Retorno da Classe EllipticCurve (Criptografia).....	75
Figura 30 - Retorno da Classe EllipticCurve (Descriptografia).....	76

Lista de Tabelas

Tabela 1 - Comparação VANET vs Redes sem fio tradicional (MENEGUETTE, 2009)	14
Tabela 2 - Comparativo de TESLA, TESLA++.	25
Tabela 3 - Exemplo de sistemas de chave pública (BARBOSA, 2003).	30
Tabela 4 - Demonstrativo do Cálculo da adição e multiplicação.....	32
Tabela 5 - Dados retirados da fórmula pelos Scilab.	33
Tabela 6 - Valores do passo intermediários do Algoritmo, aplicado ao exemplo.	39

Lista de Abreviaturas e Siglas

ASCII - American Standard Code for Information Interchange

C2P2 - Car-to-Car Peer-to-Peer

DLS - Discrete Logarithm System

DSA - Digital Signature Algorithm

DSRC - Dedicated Short Range Communications

ECC - Criptografia Com Curvas Elípticas

ECDLS - Elliptic Curve Discrete Logarithm System

ECDSA - Elliptic Curve Digital Signature Algorithm

FCC - Federal Communications Commission

ID - ID-based cryptosystem

IFS - Integer Factorization Systems

ITS - Intelligent Transport Systems

KGC - Centro de Geração de Chave

LCR - Lista de Certificados Revogados

MANET - Mobile Ad Hoc Network

NIST - National Institute of Standards and Technology

OFDM - Orthogonal frequency-division multiplexing

PKI - Public Key Infrastructure

PKG - Private Key Generator

RTA - Regional Transportation Authority

TESLA - Timed Efficient Stream Loss-tolerant Authentication

TSVC - TESLA based Secure Vehicular Communication

VANET - Vehicular Ad Hoc Network

VITP - Vehicular Information Transfer Protocol

1. Introdução

Os avanços constantes da tecnologia, a facilidade da troca de informação em todas as áreas conhecidas e equipamentos cada vez mais sofisticados apresentam um cenário onde se torna imprescindível uma infraestrutura que proporcione comunicação entre os veículos que circulam nas cidades e rodovias, para haver uma maior segurança no tráfego.

Os noticiários mostram diariamente acidentes e congestionamentos que poderiam ser evitados com a utilização de tecnologias integradas a sistemas de segurança, ou seja, uso de técnicas e equipamentos específicos que fazem a troca de informações entre os veículos, tornando possível evitar acidentes, redirecionar o trânsito em casos de congestionamentos e alertar na ocorrência de qualquer imprevisto que possa atrapalhar o fluxo nas estradas.

VANET (*Vehicular Ad Hoc NETwork*) é uma tecnologia já existente como uma subclasse da rede *Ad Hoc*. Tem como objetivo a comunicação entre veículos e/ou entre veículos e uma infraestrutura de acostamento tendo a finalidade de proporcionar segurança e autonomia nas vias rodoviárias.

Estudos mostram que essa ferramenta pode ser utilizada como controladora de congestionamentos e como alerta de possíveis colisões, entre outras funcionalidades. Seu uso poderia ser aplicado, por exemplo, no caso de uma ambulância que necessita do fluxo livre: a troca de informações emitiria um alerta para os veículos liberarem a faixa da esquerda para o deslocamento livre do veículo, possibilitando à ambulância transitar em alta velocidade, ou mesmo na ocorrência de congestionamento, direcionar a ambulância para vias secundárias onde o fluxo possa estar mais rápido. Este conceito é relativamente novo e há muito a ser estudado.

Um ponto avaliado nos estudos sobre VANETs é que a troca de informação ainda é frágil e cada vez mais fácil de ser interceptada por pessoas mal intencionadas. Em virtude desta fragilidade, os pesquisadores continuam a abordar os tópicos de segurança e troca de dados em suas pesquisas, com o intuito de minimizar as invasões e interceptações das informações que trafegam nesse tipo de rede.

Um dos protocolos utilizados para tentar solucionar estes problemas é o protocolo TESLA (*Timed Efficient Stream Loss-Tolerant Authentication*). O TESLA é um protocolo de segurança utilizado na troca de informações nas comunicações entre veículos em uma VANET. Seu funcionamento é caracterizado pelo envio da mensagem criptografada utilizando somente o valor MAC (Código Autenticação de Mensagem) único para o receptor e quando este recebe a mensagem, em seguida guarda na memória para posteriormente receber a chave MAC necessária para descriptografar a mensagem. Este protocolo foi e continua sendo estudado. No entanto, com o aumento de tipos de ataques de pessoas mal intencionadas a essas redes, surgiu a necessidade de melhorar o TESLA.

O objetivo deste trabalho é modelar uma nova implementação do protocolo TESLA, apresentando uma sugestão de alteração que tornará possível a escolha do tipo de criptografia que será utilizada no protocolo. Para esta modelagem é utilizada Criptografia de Curvas Elípticas – ECC. Além da modelagem, foi implementado o trecho do protocolo TESLA envolvendo a ECC.

O cenário que este trabalho trata envolve a comunicação entre os veículos e entre veículos e infraestrutura para melhorar a segurança nas estradas (e ruas). Não é objetivo deste trabalho abordar a segurança dos dispositivos com acesso à Internet que porventura estejam dentro do veículo (comércio eletrônico, etc.). Sendo assim, serão tratadas as comunicações autônomas que não têm o envolvimento dos usuários dos veículos.

O desenvolvimento da modelagem proposta é de alta complexidade e exige considerar detalhes específicos tanto da rede VANET, quanto do protocolo TESLA e da criptografia ECC. Neste trabalho são apresentadas de duas formas a modelagem para a implementação: uma apresentada em UML e a outra apresentada em fluxograma.

Além das modelagens em UML e fluxograma foi desenvolvida a criptografia das ECC em linguagem de programação C++, com a ferramenta Visual Studio 2010 Ultimate, com o Microsoft.Net Framework 4 em ambiente Windons XP Professional com Service Pack 3 e com hardware Intel Core 2 Duo T7250 2GHz com 3 GB de RAM.

1.1 Motivação e Justificativas

O crescente número de carros nas rodovias e nas grandes cidades metropolitanas, o aumento de possíveis ataques de pessoas mal intencionadas, o trânsito cada vez mais precário e a falta de segurança entre a comunicação entre os veículos e/ou entre os veículos e infraestrutura motivaram este trabalho.

Alguns desses problemas são tratados em trabalhos já publicados sobre as redes VANETs; Embora as VANETs já existam há algum tempo, identificou-se a necessidade de melhorá-la, uma vez que a forma de segurança da comunicação utilizada por ela são criptografias que já foram decifradas. Muitos são os motivos para essas alterações, tais como: informações falsas transmitidas por nós maliciosos, transmissão de posicionamento adulterado, divulgação de identidade de outros veículos, negação de serviço e personificação, entre outros.

Para ser possível a modelagem de um protocolo de segurança para comunicação em redes veiculares foi necessário o estudo dos conceitos envolvidos e das falhas já identificadas neste tipo de rede. Além disso, para a elaboração da modelagem tornou-se imprescindível estudar também os conceitos e características específicas do protocolo TESLA, bem como estudar o cálculo da criptografia das ECC juntamente com as ferramentas necessárias para trabalhar na referida modelagem.

Pelo grande aumento do número desproporcional dos veículos nas cidades e rodovias e a indisciplina dos motoristas, causando assim inúmeros acidentes e deixando o trânsito cada vez mais perigoso, fez surgir à necessidade de um estudo mais amplo para aumentar a segurança e diminuir o tráfego. Isso tornou-se a motivação para estudar todos estes conceitos e propor esta nova modelagem para as redes VANETs.

Alguns estudos relacionados que motivaram esta pesquisa como Meneguette (2009) e Raya e Hubaux (2005) que estudam as redes VANETs, Studer et al. (2008) que estuda o protocolo TESLA e Schneier (1996), Stallings (2008) e Boneh e Franklin (2001) que estudam a segurança das criptografias e a criptografia das ECC.

1.2 Objetivo

O principal objetivo deste trabalho de pesquisa é modelar um protocolo de segurança para comunicação em redes veiculares utilizando a modelagem UML e a modelagem fluxograma. Para atingir este objetivo foi necessário:

- Estudar as redes VANETs;
- Estudar o protocolo TESLA;
- Estudar as criptografias com foco na ECC;
- Modelar a alteração do método de comunicação das redes VANETs;
- Modelar a alteração na criptografia do protocolo TESLA;
- Implementar a encriptação e descriptação do ECC;

1.3 Organização do texto

No capítulo 2 são apresentados os conceitos, a história das redes *Ad Hoc*, sua aplicabilidade e suas características. Ainda neste capítulo são apresentados os modelos das redes VANETs, seu conceito e características, suas principais vulnerabilidades a ataques, algumas ferramentas de simulação existentes utilizadas por pesquisadores e sua segurança. Neste capítulo também são apresentados o protocolo TESLA, suas características e seu funcionamento, e o protocolo TESLA++, que é uma derivação do protocolo TESLA, e suas diferenças.

No capítulo 3 é apresentado o conceito de criptografia e suas características, a definição da criptografia ECC, seu método de desenvolvimento e seu cálculo, bem como as diferenças entre elas. E em seguida é descrito o Logaritmo Discreto que é o método para descriptografar as ECC.

No capítulo 4 é apresentada a modelagem do protocolo de segurança proposto para comunicação em redes veiculares, seus diagramas demonstrados em UML e no formato de fluxograma. Ainda neste capítulo é apresentado o desenvolvimento que foi realizado durante este trabalho e possíveis trabalhos futuros.

No capítulo 5 são apresentadas as considerações finais desse trabalho.

2. Redes sem fio *Ad Hoc*

2.1 Considerações iniciais

Na atualidade é difícil encontrar um equipamento que não esteja ligado de alguma forma a uma rede, seja por fio ou por sem fio, pois a evolução dos equipamentos e a informação rápida são cruciais para todas as áreas conhecidas, exemplo são os celulares que acabam sendo computadores de bolso, com todas as funcionalidades de um computador.

Entretanto, as redes são por muitas pessoas desconhecidas, por não trabalharem com informática ou por não ter a necessidade de aprender o mesmo, mas isso não faz com que eles não utilizem todos os dias equipamentos com estas funcionalidades.

Para os que sabem e principalmente para os que desconhecem neste capítulo, será apresentado o conceito das redes sem fio *Ad Hoc* e suas aplicações, vantagens e desvantagens. Será apresentado também o conceito das redes VANETs, seus modelos e protocolos, seus principais tipos de atacantes e seus ataques, características da rede e suas ferramentas para simulação.

Ainda neste capítulo será apresentado o protocolo de segurança TESLA e uma derivação do mesmo, TESLA++.

2.2 Redes *Ad Hoc*

As redes *Ad Hoc* surgiram em meados da década de 70 em pesquisas nos Estados Unidos, quando estudavam como poderia ser a comunicação via rádio em um ambiente tático

militar, onde existia grande necessidade de mobilidade dos dispositivos que se comunicavam através de uma rede.

As redes *Ad Hoc* são também conhecidas como MANET (*Mobile Ad Hoc Network*), onde os dispositivos são capazes de trocar informações diretamente entre si. Esse tipo de redes não têm necessidade de infraestrutura, pois é formada por nós móveis que podem funcionar tanto como roteadores ou receptores.

Ao contrário do que ocorre em redes convencionais, não há pontos de acesso, ou seja, não existem estações de suporte à mobilidade, não há infraestrutura de conexão. Por não ter esta necessidade de infraestrutura, permite grande flexibilidade e uso em lugares hostis, como ambientes de guerra ou catástrofes. Por esse motivo, redes *Ad Hoc* são indicadas principalmente em situações onde não se pode, ou não faz sentido, instalar uma rede fixa.

Com esta facilidade de locomoção nas redes *Ad Hoc*, os seus nós podem se mover arbitrariamente, e conforme a necessidade, a topologia da rede se adapta e reconfigura as melhores rotas.

Considerando o fato dos nós das redes *Ad Hoc* não estarem ligados a uma infraestrutura, existem algumas limitações como o tamanho da banda, o tempo da energia das baterias dos equipamentos e o roteamento.

2.3 Vantagens e Desvantagens

Apesar da facilidade da utilização das redes *Ad hoc* é preciso avaliar suas vantagens e desvantagens em comparação com as redes com infraestrutura, redes fixas que utilizam cabeamento estruturado.

Algumas das vantagens identificadas são: I - a instalação rápida por não haver a necessidade de uma infraestrutura; II - tolerância a falhas por constantemente se adaptar e reconfigurar as rotas, permitindo assim que perdas de conectividade entre os nós possam ser facilmente resolvidas; III - conectividade direta entre os nós, desde que esteja dentro do alcance do outro e a mobilidade, vantagem primordial em relação às redes fixas.

Por outro lado apresentam algumas desvantagens como: I - o roteamento, pois a mobilidade dos nós e a topologia de rede dinâmica contribuem diretamente para tornar a construção de algoritmos de roteamento um dos principais desafios em redes *Ad Hoc*; II - a localização, pois não existe a ligação do endereço da máquina com a posição atual do nó, e também não há informações geográficas que auxiliem na determinação do posicionamento desse nó; III - taxas de erros, pois existe um aumento dos enlaces em comparação com as

redes estruturadas; IV - tamanho da banda, pois existe a limitação de 2Mbps em uma rede *Ad Hoc*, sendo que nas redes estruturadas pode chegar a 1Gbps; V - e energia, pois os equipamentos são limitados ao tempo da duração da energia interna.

2.4 Redes veiculares *Ad Hoc* – VANET

VANET é um conceito de redes *Ad Hoc* aplicadas a veículos, que proporciona maior segurança e autonomia nas vias rodoviárias. Este conceito está sendo estudado há algum tempo e Meneguette (2009) e Sichitiu e Kihl (2008, apud MENEGUETTE, 2009, pag. 30) diz que as redes VANETs podem atuar também como controladoras de congestionamentos e como alerta de possíveis colisões, entre outras funcionalidades. Esta comunicação é feita de veículo para veículo ou de veículo para estrada, e pode se dar através de torres de comunicação telefônicas ou postes de acesso externo.

Uma evolução na área automobilística é a ideia de “consciência”, que significa que um veículo é consciente de sua vizinhança, inclusive da presença e localização de outros veículos. Os veículos conscientes possuem uma rede de sensores conectados a uma central de processamento de dados que provê interfaces de comunicação baseadas nos padrões Ethernet, USB, Bluetooth, 802.11, dentre outros (IEEE SECURITY AND PRIVACY, 2004).

Neste cenário, um conjunto de veículos que se comunicam entre si é um exemplo de rede móvel sem fio, inserida em um contexto de computação ubíqua. Neste caso, além dos sistemas implementarem a comunicação entre esses veículos, também precisam proteger as informações pessoais de seus usuários (IEEE SECURITY AND PRIVACY, 2004).

As redes *Ad Hoc* tiveram um grande crescimento desde meados da década de 90, quando começaram pesquisar com foco em redes sem fio que apresentaram formulações de vários protocolos, atingindo o topo em experimentos consequentemente de agora em diante aplicados também em redes *Ad Hoc* veiculares.

As tecnologias que estão sendo desenvolvidas para as VANETs focam principalmente na comunicação entre veículo-veículo e veículo-estrada, conforme mostra a Figura 1, que estabelece uma comunicação de forma a melhorar e coordenar o fluxo de veículos nas vias, pois a maioria dos nós são veículos e é necessário lembrar que a troca de dados entre os veículos tem que ser de forma rápida, considerando que a velocidade nas rodovias costuma ser acima de 100 Km/h e os veículos se deslocam para todas as direções (MENEGUETTE, 2009).

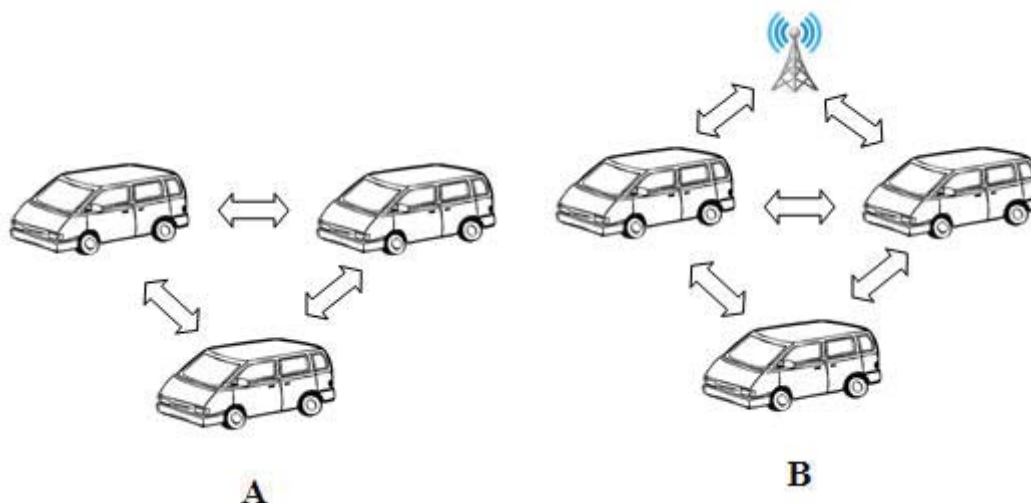


Figura 1 - Comunicação de redes *Ad Hoc* veiculares (A) Troca direta de informações entre veículos. (B) Troca de informações entre veículos por meio de infraestrutura.

A Figura 1 ilustra a comunicação entre veículos em redes VANETs, onde a comunicação entre veículos pode ser direta (Figura 1A) ou por meio de uma infraestrutura de acostamento, que possibilitará inclusive acesso à internet aos veículos (Figura 1B) (MENEGUETTE, 2009).

2.4.1 Características de uma rede veicular

Segundo Meneguette (2009) as redes VANETs apresentam diversas características importantes e necessárias para uma rede móvel, dentre elas: serem auto organizáveis, ter alta taxa de mobilidade, ter grande velocidade de transmissão, topologia, não ter necessidade de racionamento de energia, largura de banda, fragmentação da rede, ter comunicação direta e ausência de taxa de serviço.

O desenvolvimento de aplicações ou serviços para VANETs tem suas próprias especificações conforme citado por Wisilpongphan, et al. (2007, apud MENEGUETTE, 2009, p. 31).

...faz uso de técnicas próprias ou modificadas das redes sem fio, em virtude das diferenças entre as redes sem fio convencionais. As redes VANETs sofrem rápidas mudanças em suas conexões e também têm que lidar com diferentes tipos de densidades das redes.

Comparando as redes MANETs e as redes VANETs, as redes VANETs têm uma constante perda de conectividade por falta de equipamento nas rodovias ou pelo tráfego

denso, mas diferente das redes MANETs que ficam sem uso devido à falta de um transmissor, necessário na mesma, item este que não é necessário na VANET, pois a comunicação pode ser de veículo para veículo. Outra característica marcante que diferencia as redes VANETs é não ter limite de uso computacional e nem limite de energia, pois não são limitados a pequenas baterias como, por exemplo, os celulares nem processadores de baixo processamento como as redes MANETs.

A Tabela 1 apresenta um resumo da comparação entre a rede VANET e a rede sem fio tradicional.

Tabela 1 - Comparação VANET vs Redes sem fio tradicional (MENEGUETTE, 2009)

Características	VANET	Sem Fio Tradicional
Auto configurável	Sim	Sim
Mobilidade	Rápida, com Restrições	Relativamente baixa, Sem restrição
Transmissão	Rápida, Densidades diferentes	Relativamente baixa, Densidades diferentes
Topologia	Dinâmica	Dinâmica
Energia	Constante	Depende de bateria
Banda	Restrita	Restrita
Fragmentação	Frequente	Aleatória

É possível observar na Tabela 1 que há diferenças significativas entre a rede VANET e a rede sem fio tradicional. As principais são: a velocidade que os nós (veículos) podem ter na rede, o curto tempo de transmissão que os nós têm para transmitir e a frequente fragmentação da rede (MENEGUETTE, 2009).

Numa rápida comparação das redes de celulares e as redes veiculares observa-se que:

- **Há Comunicação Direta:** a comunicação pode ser direta com outro nó (veículo) na rede, não necessitando de uma estação para transmissão como um satélite ou outros dispositivos quaisquer. Sendo assim, quase não existe atraso na transmissão dos dados, fazendo com que seja mais rápida. Outra característica importante é que os nós (veículos) podem se comunicar em áreas que as redes de celulares não cobrem como em áreas rurais.
- **Há Ausência de Taxa de Serviço:** as redes VANETs não tem a necessidade da utilização de uma operadora como nas redes MANETs, com isso não exige

infraestrutura para a comunicação, podendo assim estar em qualquer lugar e se comunicar apenas entre os nós e pistas.

2.4.2 Modelo de uma VANET

Os nós de comunicação em uma VANET são compostos por veículos e estações base que compõem a infraestrutura da rodovia ou rua. Veículos podem ser particulares (pertencentes a uma pessoa/empresa) ou públicos (pertencentes a organizações públicas como, por exemplo, os ônibus de transporte urbano). As estações base podem pertencer a órgãos públicos ou concessionárias que administram tais infraestruturas.

Como a maioria dos nós da rede é constituída por veículos, a dinâmica da rede consistirá de uma mobilidade permanente. Como consequência, o tempo de conexão entre vizinhos é curto, uma vez que em rodovias a velocidade média dos veículos é acima de 100Km/h, com velocidades relativas acima de 200Km/h. Outro aspecto da dinâmica da rede é que as trajetórias dos veículos são definidas pelo traçado das rodovias/ruas.

O tamanho de uma VANET é uma característica importante. Apesar de possuir centenas de milhões de nós distribuídos por todos os continentes, a comunicação nessa rede será local, permitindo o particionamento da rede em sub-redes menores e totalmente escaláveis.

Uma vantagem de uma VANET sobre uma rede *Ad Hoc* convencional é que os veículos não estão limitados em termos de poder computacional e consumo de energia. Um veículo poderá ter dezenas ou centenas de microprocessadores interligados, cada um responsável pelo processamento de dados específicos.

Segundo Raya e Hubaux (2005), as aplicações que serão desenvolvidas para uma VANET podem ser divididas em duas categorias: aplicações relacionadas a segurança e outras aplicações:

- **Aplicações relacionadas a segurança:** aplicações para evitar colisões, direção cooperativa e otimização de tráfego são aplicações que se encaixam nessa categoria. A característica comum às aplicações dessa categoria é a preocupação com as situações onde uma falha no sistema pode causar danos à vida de seres humanos. Assim, o perfeito funcionamento das aplicações nessa categoria é fundamental, independentemente das possíveis ameaças introduzidas por um adversário.

- **Outras aplicações:** estão incluídas nessa categoria as aplicações relacionadas ao pagamento de serviços (ligações telefônicas, por exemplo), serviços de localização (de um posto de combustível, por exemplo), entretenimento (acesso a internet, por exemplo), dentre outros. Logicamente, alguns desses serviços demandam segurança, mas por não envolver diretamente o risco à vida humana, são classificados nessa categoria. Além do mais, o problema de segurança dessas aplicações é tratado diretamente pelos sites e pelas aplicações.

2.4.2.1 Mensagens de Segurança

Apenas as aplicações relacionadas a segurança são consideradas neste trabalho por que a falha no sistema pode causar danos à vida de seres humanos. Nesse contexto, podem-se classificar as mensagens em três classes baseadas nas suas propriedades relacionadas à privacidade e necessidades de tempo real: mensagens sobre informações de tráfego, mensagens gerais sobre segurança e mensagens de responsabilidade (RAYA E HUBAUX, 2005).

- **Mensagens sobre Informações de Tráfego:** são usadas para disseminar as condições de tráfego em uma determinada região portanto, afeta a segurança dos usuários apenas de forma indireta, prevenindo acidentes em potencial devido ao congestionamento de veículos.
- **Mensagens Gerais de Segurança:** são usadas por aplicações públicas de segurança, como por exemplo, direção cooperativa para evitar colisões. Tais aplicações são bastante sensíveis ao tempo, necessitando ser enviadas dentro de intervalos de tempo curtos para que sejam efetivas.
- **Mensagens de Responsabilidade:** distinguem-se das outras por envolver troca em situações de responsabilidade do emissor tais como em acidentes, catástrofes naturais, etc. Portanto, a responsabilidade do emissor da mensagem deve ser determinada revelando-se sua identidade às autoridades competentes.

Uma propriedade comum em todas as classes de mensagens é que cada mensagem chega ao seu destino através de *broadcast* e geralmente não possuem relação com outras mensagens anteriores. O conteúdo de uma Mensagem Geral de Segurança típica inclui a posição, velocidade, direção e aceleração do veículo, incluindo mensagens específicas sobre eventos de tráfego (notificação de congestionamento ou acidente, por exemplo).

Caso o emissor encontre uma situação anormal, um acidente, por exemplo, esses dados ajudam os receptores a calcular suas posições em relação ao emissor e determinar se eles estão em perigo. Nesse caso, as mensagens não necessariamente vêm acompanhadas do identificador do emissor.

Uma característica importante em redes *Ad Hoc* é o *multihopping*. Mas, devido às características de *broadcast* do protocolo 802.11p (protocolo que está sendo desenvolvido pela IEEE como padrão para comunicação interveicular), as mensagens gerais de segurança são transmitidas sobre *singlehop* com potência de sinal suficientemente grande para chegar a veículos dentro de um raio considerável, eliminando a necessidade de *multihop*.

Porém, alguma forma de *multihop* ainda pode ser implementada, caso um veículo receba uma mensagem e julgue que o conteúdo daquela mensagem poderia afetar a segurança de outros veículos trafegando mais atrás, esse veículo poderia reenviar a mensagem, avisando aos outros veículos dos problemas à frente.

2.4.3 Protocolo para Mensagens de Segurança

Em linhas gerais, um protocolo para mensagens de segurança deverá possuir algumas características (RAYA; HUBAUX, 2005):

- De acordo com as especificações do IEEE 802.11p, assume-se que cada veículo V envia mensagens periódicas do tipo *singlehop* a cada 300ms com um raio de alcance de 110m a 300m;
- O intervalo entre cada mensagem cai para 100ms e o raio para 15m se o veículo estiver parado ou a uma velocidade baixa (menor que 16Km/h, por exemplo);
- Cada veículo toma decisões com base nas mensagens recebidas, podendo transmitir mensagens novas relacionadas a essas mensagens. Por exemplo, se V recebe um aviso de emergência de outro veículo W e com base em suas posições relativas determina que V também está em perigo, V envia suas próprias mensagens de aviso.

2.4.4 Ataques em Redes Veiculares

Como não é possível prever todos os tipos de ataques que uma rede VANET estará sujeita no futuro, em linhas gerais essa seção apresenta algumas considerações sobre um possível atacante.

Tais considerações ajudarão a nortear possíveis medidas que poderão ser tomadas para eliminar por completo ou então conter alguns dos ataques mais prováveis de serem lançados por um atacante sobre uma rede VANET.

2.4.4.1 Modelo de Atacante

Para classificar as capacidades de um atacante, Raya e Hubaux (2005) definem três dimensões:

Membros Internos versus Membros Externos: Membros Internos são membros autenticados dentro da rede capazes de se comunicar com outros membros e Membros Externos, como não são autenticados na rede, são considerados intrusos e, portanto possuem uma capacidade limitada de atacar Membros Internos.

Adversários mal Intencionados versus Adversários Racionais: Adversários mal Intencionados não procuram benefício pessoal com um ataque, buscam apenas prejudicar os usuários ou a própria rede com o ataque. Já os Adversários Racionais utilizam-se de ataques para obter benefício próprio. Esses tipos de ataques são mais previsíveis se comparados aos ataques por adversários mal intencionados.

Adversários Ativos versus Adversários Passivos: Um Adversário Ativo pode gerar pacotes ou sinais de forma a espionar a rede, enquanto que um Adversário Passivo não emite sinal algum. Fica apenas ouvindo a rede para captar informações sigilosas de seus usuários.

2.4.4.2 Possíveis Ataques

Nessa seção serão apresentados possíveis ataques que poderão ser feitos por adversários sobre as mensagens que trafegam sobre uma rede veicular segundo Raya e Hubaux (2005).

Informações Falsas: adversários podem transmitir informações falsas na rede para afetar o comportamento de outros usuários, um exemplo seria enviar informações falsas a respeito das condições de tráfego da via numa tentativa de liberá-la para uso exclusivo.

Transmissão de Posicionamento Adulterado: adversários podem usar este ataque para deliberadamente alterar suas posições, velocidades, direções, etc. (para se esquivar da responsabilidade em caso de acidente, por exemplo).

Divulgação de Identidade de outros Veículos: adversários podem usar este ataque para rastrear a posição de outros veículos.

Negação de Serviço: um atacante pode ter a intenção de tirar a rede do ar, ou mesmo causar algum acidente. Exemplos desse tipo de ataque são: interferência no canal e envio agressivo de mensagens falsas.

Personificação: usando identidades falsas, o adversário tem um comportamento ativo e finge ser outro veículo.

2.4.4.3 Requisitos de Segurança em Redes VANET

Essa seção apresenta alguns dos requisitos gerais de segurança que deverão ser implementados em redes VANET como defesa aos ataques apresentados na seção anterior (RAYA E HUBAUX, 2005):

- **Autenticação:** as reações dos veículos a eventos deverão ser baseadas em mensagens legítimas geradas por membros legítimos dessa forma deverá existir algum mecanismo que possibilite autenticar cada membro e relacioná-lo com cada mensagem enviada por ele.
- **Verificação da Consistência de Dados:** a legitimidade de uma mensagem engloba também sua consistência com mensagens similares, que são aquelas geradas em intervalos de tempo curtos, nas proximidades de um determinado local. Esse requisito é necessário já que um veículo pode ser autenticado, mas gerar dados inconsistentes.
- **Não-Repúdio:** usuários que causem um acidente devem ser identificados. Assim, qualquer usuário não deve ser capaz de impedir a transmissão de uma mensagem, principalmente as mensagens de responsabilidade.

- **Privacidade:** a privacidade de veículos contra observadores não autorizados deve ser garantida.
- **Disponibilidade:** mesmo assumindo um canal de comunicação robusto, alguns ataques podem interromper a comunicação na rede. Assim, a comunicação na rede deverá prever disponibilidade através de outros meios.

2.4.5 Ferramentas para Simular VANET

Os simuladores são ferramentas que permitem aos desenvolvedores a realização de simulações sem a utilização de grandes gastos com infraestrutura física, tornando testes em projetos viáveis antes de seu desenvolvimento.

Existem diversos simuladores para n ferramentas e métodos, mas não existe um simulador específico para as redes VANET. As simulações são realizadas utilizando-se de simuladores para redes móveis em conjunto com simuladores de movimento.

Alguns exemplos de possíveis simuladores que podem ser utilizados em redes móveis. São eles:

- NS-2 (*Network Simulator version 2*) é largamente utilizado na comunidade para simulação de redes por trabalhar ao nível de pacote, com protocolos de transporte (TCP), protocolos de controle de acessos, redes *Ad Hoc*, protocolos *Multicast* e protocolo para comunicação via satélites (FALL E VARADHAN, 2002);
- GloMoSim (*Global Mobile Information System Simulator*) é um simulador de rede sem fio que foi desenvolvido utilizando a capacidade de eventos discretos fornecidas pelo Parsec, que utilizada uma linguagem de programação paralela. (BAJAJ et al, 1999);
- MOVE (*MObility model generator for VEhicular networks*) é uma ferramenta desenvolvida em Java e foi desenvolvida para facilitar e criar modelos realistas de mobilidade para simulações de VANET, para isso são utilizados dois editores, um de mapas de estradas e um de movimentação de veículos. O resultado desta simulação é um mapa das movimentações dos veículos, estes mapas criam alguns arquivos de trajetória que são posteriormente utilizados por outros simuladores (ns-2, por exemplo) (KARNADI et al, 2007);

- SUMO (*Simulation of Urban Mobility*) é um simulador de tráfego urbano utilizado na simulação de redes veiculares. O SUMO simula a movimentação de diferentes tipos de veículos sem a ocorrência de colisões, suportando vias de múltiplas faixas de rolamento. O funcionamento do SUMO baseia-se em um modelo de comportamento dos motoristas combinado com controladores de tráfego presentes nas vias (KRAJZEWICZ et al, 2002);
- TraNS (*Traffic and Network Simulation Environment*) em resumo esta ferramenta é a utilização de dois outros simuladores o ns-2 e o SUMO, que pode utilizar modelos de mobilidade realistas e verificar a influência do comportamento do tráfego, com base na comunicação entre os veículos (PIORKOWSKI et al, 2008);
- TraCI (*Traffic Control Interface*) que faz a ligação entre o simulador de redes e o simulador de tráfego. O TraCI envia comandos de mobilidade para o simulador de tráfego podendo realimentar o simulador de redes com novos modelos de mobilidade.
- VanetMobiSim embora seja uma extensão do CanuMobiSim que tem seu foco em simular redes moveis, o VanetMobiSim tem como foco tentar simular mobilidade veicular e esta ferramenta está dividida em dois níveis: **macroscópica** que pode importar mapas de US Census Bureau TIGER, e além disso, ter suporte para pistas multi-estradas, respeita fluxos direcionais, restrições de velocidade e sinais de trânsito nos cruzamentos, e **microscópica** quando fornece o realismo da comunicação entre carro para carro e carro para infraestrutura. Neste caso, os veículos regulam sua velocidade dependendo dos carros nas proximidades, e respeitam a ultrapassagem e sinais de transito em cruzamentos.

Durante os estudos deste trabalho foram testados os simuladores NS-2 e VanetMobiSim, mas não se obteve sucesso, seja pela dificuldade de configuração da ferramenta ou por não conseguir simular uma rede VANET. Mas como o simulador não é o foco deste trabalho não serão abordados detalhes individuais.

2.4.6 Segurança em Redes Veiculares

A segurança na troca de dados nas redes VANETs é importante, pois visa sempre possibilitar a troca de informações nas estradas com dados confiáveis e autênticos, possibilitando um planejamento de rotas rodoviárias e comércios eletrônicos. Para estas aplicações em particular, a segurança das redes são ainda mais fundamentais, na verdade são questões chaves para a confiança.

Com estas necessidades de seguranças, a preservação da privacidade do indivíduo é outra questão muito importante para redes VANETs. Os nós (carros/motoristas) não querem que suas informações pessoais sejam reveladas (nome, rota, movimento e dados do usuário) e este problema está um pouco longe de ser resolvido por completo, pois por se tratar de rodovias ou cidades, os usuários não estarem constantemente parados e os sistemas serem onipresentes, os mesmos podem sofrer ataques de qualquer lugar ou ter perda de dados.

Nas aplicações, tais como prevenção de colisões e de tráfego em geral, o usuário deve enviar mensagens relacionadas a esses serviços a outros veículos ou dispositivos da estrada, o que viola a privacidade do usuário, independentemente da sua intenção. No entanto, no caso de acidentes de carro ou em crimes, as informações da identidade devem ser reveladas para as autoridades contribuindo assim para a captura do infrator.

2.4.6.1 Relações de Confiança

A relação de confiança é uma forma antiga de se determinar se é seguro ou não certo indivíduo, e não é diferente nas redes. Mas as autenticações cotidianas utilizadas pelos usuários são muito comuns para a segurança em comunicação e nos casos de computação ubíqua, onde existem inúmeros elementos desconhecidos e independentes, este controle se torna ainda pior podendo ser até prejudicial a rede.

A Confiança nas redes se define de forma que um nó acredita, espera e aceita que o outro nó seja confiável, e desta forma ambos têm benefícios mútuos. Nesses casos, um sistema baseado em reputação pode rastrear o comportamento de nós separando-os em nós bem comportados e nós com comportamento suspeito, atribuindo a cada nó um grau de confiabilidade.

Este grau de confiabilidade define o quanto este nó é seguro em qualquer interação com outro nó. Cada nó define o grau de confiança perante um segundo nó, assim sendo o mesmo nó pode ter diversos graus de confiabilidade para diversos nós diferentes.

Diversos trabalhos tratam desse assunto, apresentando técnicas e discutindo a aplicabilidade de um sistema baseado em Reputação para implementar segurança em redes *Ad Hoc* (BOUKERCHE et al, 2009; YONGLIN; BOUKERCHE, 2008).

2.4.6.2 TESLA

O TESLA (*Timed Efficient Stream Loss-tolerant Authentication*) é um protocolo de segurança utilizado na troca de informações nas comunicações e baseia-se na perda de tempo de sincronização (tempo que leva para a mensagem chegar ao destino) entre o emissor e o receptor e apesar de usar exclusivamente funções de criptografia simétrica MAC (Código de Autenticação de Mensagens), esse protocolo atinge propriedades assimétricas (RUAN; HORI, 2012; STUDER et al, 2008).

Seu funcionamento é caracterizado pelo envio da mensagem criptografada utilizando somente o valor MAC único para o receptor e quando este recebe a mensagem, em seguida guarda na memória para posteriormente receber a chave MAC, necessária para a descriptografar a mensagem. A autenticação da mensagem é por difusão (*broadcast*). Em comunicações ponto-a-ponto, a utilização de algoritmos MAC para garantir a autenticação das mensagens é simples, porém, em comunicações em difusão, os diversos destinatários teriam que conhecer a chave MAC, por isso, a autenticação em difusão necessita de primitivas assimétricas.

É possível diferenciar o protocolo TESLA dos protocolos assimétricos tradicionais, como RSA, pois a assimetria é obtida através de sincronização de relógios e atraso na divulgação da chave, ao invés de realizar operações que exigem grande poder computacional.

Segundo STUDER et al. (2008) pode-se autenticar uma mensagem M , enviando a mensagem e um MAC para o receptor, utilizando um pacote com a chave do remetente para este intervalo (k_i).

Os destinatários salvam a mensagem inteira até que o remetente transmita a chave MAC. Após o período de divulgação da chave, o remetente transmite a mesma.

Para autenticar a mensagem, se faz necessário uma verificação se existe a mensagem correspondente armazenada no receptor daquela chave MAC, caso encontrado mensagem autentica e descriptografada.

No entanto, o protocolo TESLA tem alguns pontos a serem melhorados, por exemplo, o receptor aloca memória para guardar a mensagem enviada do remetente até o momento da chegada do MAC, isso requer uma grande quantidade de espaço de memória.

Diante desta necessidade muitos invasores maliciosos tentam sobrecarregar a memória do receptor (ataque de poluição) para deixar lenta a rede ou para derrubar a mesma.

Este problema de memória foi solucionado por STUDER et al. (2008) que propôs o TESLA++ que será descrito a seguir.

Na Figura 2 é mostrado como é enviado a mensagem para o destinatário e logo após a chegada da mesma é enviado o MAC, necessário para descriptografar.



Figura 2 - Representação do envio da informação e da Mac para o receptor no TESLA.

2.4.6.3 TESLA++

Para solucionar o problema de ataques de memória DoS no protocolo TESLA foi proposto o TESLA++, que é uma versão modificada do TESLA proposto pelo Studer et al. (2008), e esta nova versão reduz os requisitos de memória do receptor sem sacrificar a segurança e fornece autenticação *broadcast* usando criptografia simétrica, assim, o receptor armazena apenas um MAC autogerado para reduzir os requisitos de memória. Os receptores armazenam somente uma versão abreviada dos dados do remetente, que transmite primeiro o MAC e depois a mensagem correspondente.

Conforme Studer et al. (2008) pode-se autenticar a mensagem M em TESLA++, quando o remetente transmite primeiro o MAC $mac_s = mac_{ki}(M)$, que é calculado com a chave k_i atual, juntamente com o índice de chave i . Após a recepção, usando o índice de chave i e o tempo associado com o início da cadeia de chave do remetente, o destinatário primeiro verifica a condição da segurança para garantir que a chave k_i ainda não foi transmitida e, portanto, ainda conhecida apenas pelo remetente. Se a condição de segurança não é segura, o receptor descarta o MAC, pois um invasor poderia já ter recebido a chave k_i correspondente.

Quando há a segurança, o receptor então recebe os dados re-MACs através de um k_{Recv} chave local secreto que só é conhecido pelo receptor $MAC_R = MAC_{k_{Recv}}(MAC_S)$ e armazena este MAC MAC_R curtos juntamente com o índice de chave.

Uma vez que a chave k_i pode ser divulgada, o remetente vai transmitir qualquer mensagem e a chave usada para calcular MACs. Para verificar uma mensagem, o receptor primeiro verifica a validade de k_i , seguindo a cadeia de chave única de volta para uma chave confiável. O receptor calcula o MAC curto da mensagem e compara com o MAC armazenado na memória. Se o receptor tem uma chave MAC que corresponde ao índice da memória, o receptor considera a mensagem autêntica.

Se nenhum dos pares armazenado corresponde ao valor calculado recentemente, o receptor considera que a mensagem não é autêntica e descarta a mesma.

Mas com a utilização da o TESLA++ surgiram alguns problemas não existentes no TESLA, como a perda de pacotes, fazendo com que o MAC curto armazenado na memória do receptor nunca tenha sua mensagem entregue. A partir desta falha foi necessário promover algumas alterações na forma de limpeza da memória do receptor, desta forma, quando uma mensagem é entregue bem sucedida todos os MACs anteriores do mesmo remetente armazenado na memória são descartados.

Além da perda de pacotes, o TESLA++ utiliza o dobro do desempenho da rede em comparação com o TESLA.

Na Figura 3 mostra o envio do MAC curto para o receptor e logo após o envio da informação para ser descriptografada.



Figura 3 - Representação do envio do MAC curto e em seguida a informação para o receptor no TESLA++.

A Tabela 2 mostra algumas das principais diferenças entre TESLA e TESLA++, como, por exemplo, a necessidade de memória utilizada no TESLA e o dobro de recursos necessários para o TESLA++ em comparação com o TESLA.

Tabela 2 - Comparativo de TESLA, TESLA++.

	TESLA	TESLA++
Protocolo	Sim	Sim
Criptografia	Simétrica	Simétrica
Manda a mensagem junto com a chave	Sim	Não
Memória no receptor	Guarda a	Guarda um MAC

	mensagem e o MAC	curto
Não-repúdio	Repúdio	Repúdio
Desempenho da rede	Auto	Dobro do TESLA
Perda de Pacote	Baixo	Auto
Prevenção de ataques baseados em DoS Sobrecarga do sistema.	Sim	Sim
Suporte para multi-hop	Não	Não
Prevenção de ataques baseados em DoS Memória	Não	Sim
Autenticação	Sim	Sim

2.5 Considerações Finais

Neste capítulo foram estudadas as redes *Ad Hoc*, seus conceitos e técnicas e uma derivação da mesma, as redes VANETs que têm como principal objetivo a segurança nas rodovias e a troca de dados nas mesmas. Mas há inúmeras falhas neste processo que permitem várias formas de ataques e invasões que causam transtornos para seus usuários.

Pensando nestas falhas foram estudadas algumas formas de segurança, como por exemplo, TESLA e TESLA++, que são protocolos de segurança utilizado na troca de informações nas comunicações. Seu funcionamento é caracterizado pelo envio da mensagem criptografada utilizando somente o valor MAC, que no decorrer deste estudo serão abordadas com a rede VANET.

Nesse capítulo foram apresentados vários conceitos e características das redes *Ad Hoc* e VANETs, entre outros itens necessários para continuação da leitura deste trabalho, com isso o leitor já tem o embasamento teórico para continuar.

No próximo capítulo são abordadas formas de segurança e suas ferramentas como também algumas criptografias utilizadas e o desenvolvimento da ECC.

3. Ferramentas para Segurança

3.1 Considerações iniciais

Existem diversas ferramentas desenvolvidas para gerenciar a segurança no nosso dia-a-dia, como a senha do banco, do sistema da empresa, de um site de jogos e etc. Todas utilizam de alguma forma, não importa qual, uma criptografia para proteger seus dados e torná-los ilegíveis para um desconhecido.

Neste capítulo, é apresentado o conceito de criptografia e a ECC, que é considerada a criptografia mais segura, seu modelo e seu cálculo para o desenvolvimento e a forma de descriptografia utilizando o Logaritmo Discreto (SHOUZHI, X. et al, 2012).

3.2 Criptografia

Em poucas palavras pode-se definir a criptografia como sendo a ciência que utiliza números e expressões matemáticas para cifrar (encriptar) e decifrar (desencriptar) dados. A criptografia de um dado importante nada mais é do que guardar o mesmo de forma que apenas o destinatário consiga entendê-la, podendo ser enviada por redes de comunicação, como a internet, sem que ninguém consiga decifrá-la.

Para descobrir se uma criptografia é segura ou não pode-se utilizar a criptoanálise, que é a ciência que analisa o grau de complexidade e o tempo para descriptografar uma informação. A criptoanálise clássica envolve combinações analíticas, ferramentas matemáticas, paciência, determinação e sorte. Pode-se classificar grosseiramente as criptografias como sendo fraca ou forte, e é levado em conta o tempo e os recursos que

devem ser gastos para repor o texto original. Uma criptografia fraca pode ser facilmente reescrita na sua forma original.

Porém uma criptografia forte é extremamente difícil de quebrar sem o uso da ferramenta adequada, e mesmo com a ferramenta adequada, o tempo para recompor o texto original fica na ordem de milhões de anos. Para criptografar ou descriptografar um texto são utilizadas funções matemática que fazem uso de chaves (um número, uma palavra ou uma frase) necessárias para criptografar um texto puro.

Para cada função matemática ou chave diferente obtém-se resultados diferentes, mesmo tendo um mesmo texto puro. Para classificar se o texto criptografado é seguro depende de duas coisas: da força do algoritmo criptográfico e do secretismo da chave. Um algoritmo criptográfico com todas as possíveis chaves e todos os protocolos que o fazem funcionar compõem um criptossistema.

Existem duas formas de criptografia: a criptografia de chaves públicas e a criptografia de chaves privadas. A primeira é também conhecida por Criptografia de Chaves Assimétricas e a segunda é conhecida por Criptografia de Chaves Simétricas.

De acordo com Molva; Samfat e Tsudik (1994) existem algumas promissoras pesquisas na parte de criptografia, que eliminam a necessidade de certificados de chaves públicas. Por exemplo, Boneh e Franklin (2001) a identidade baseada em criptografia do sistema, um método de obtenção de criptografia de chave pública: nele, a chave pública de uma entidade se deriva de uma sequência de identidade original, por exemplo, um endereço de e-mail ou nome distinto qualquer. Isso pode ser um número de identificação de um veículo, no caso de VANET.

Como ainda não se utilize destas técnicas estudadas por Boneh e Franklin (2001) tem-se que lembrar que os certificados de chaves públicas contêm uma data de vencimento programada. No caso relatado por Boneh e Franklin (2001) não há esta chave, como mostra o exemplo a seguir:

Alice criptografa os e-mails para serem enviados a Bob usando a chave pública: "bob@company.comcurrent-year".

Ao fazê-lo, Bob pode usar sua chave privada apenas durante o ano corrente. Uma vez por ano Bob precisa obter uma nova chave privada do PKG (Private Key Generator). Assim, obtém-se o efeito de vencimento anual da chave privada. Nota-se que, diferentemente das PKI (Public Key Infrastructure) existentes, Alice não tem necessidade de obter um novo certificado de Bob cada vez que ele troca a sua chave privada.

Alice poderia fazer essa abordagem mais granular, criptografando e-mail para Bob, utilizando "Bob@company.comcurrent-date". Deste modo forçaria Bob a obter uma nova chave privada todos os dias.

3.2.1 Criptografia de Chave Assimétrica

Para a criptografia de chaves assimétrica são necessárias apenas duas chaves, a **pública**, conhecida por todos, e a **privada**, conhecida apenas por uma pessoa, e nesse processo uma é utilizada para cifrar e outra para decifrar as mensagens. Assim, qualquer pessoa que possua a chave pública pode enviar uma mensagem criptografada, mas apenas o destinatário que tenha a chave privada pode decifrar a mesma. Exemplos desta criptografia são: Deffie-Hellman, DSA e RSA (SCHNEIER, 1996; STALLINGS, 2008).

3.2.2 Criptografia de Chave Simétrica

Para criptografia de chave simétrica é necessário apenas uma chave que é utilizada para cifrar e decifrar a mensagem. Com isso a mensagem é obtida justamente pelo uso de uma única chave simétrica compartilhada. As mensagens cifradas pela chave simétrica somente poderão ser decifradas pela mesma chave. Exemplos de algoritmos para criptografia de chave simétrica são: Crypt, DES, DES triplo, IDEA, Lucifer, RC (RC2, RC4, RC5) e AES (SCHNEIER, 1996; STALLINGS, 2008).

3.3 Criptografia Com Curvas Elípticas (ECC)

A Criptografia de Curvas Elípticas ou ECC (das iniciais em inglês *Elliptic Curve Cryptography*) é uma variante da criptografia assimétrica ou de chave pública que é baseada na matemática das curvas elípticas sobre campos finitos. Seus criadores Victor Miller e Koblitz Neil argumentam que a ECC pode ser mais rápida e usar chaves mais curtas do que os métodos antigos como o RSA, além de proporcionar ao mesmo tempo um nível de segurança equivalente (WASHINGTON, 2008) (SAVARI et al, 2012).

As ECC são estudada há algum tempo para a segurança das informações com chave reduzida/simplificada. Com base em inúmeros estudos realizados observou-se que a criptografia com curvas elípticas é uma excelente opção para segurança das informações, por seu grau de complexidade no entendimento de seu cálculo (SHOUZHI, X. et al, 2012).

Segundo Barbosa (2003) é muito importante salientar que curvas elípticas não são elipses. O nome foi dado porque foram definidas como um objeto matemático (uma curva) que mostram uma equação cúbica, as mesmas que são usadas para calcular o comprimento de arco de uma elipse. As curvas podem assumir formatos diversos, dependendo apenas dos parâmetros utilizados. A partir de um conjunto de pontos dessa curva, é possível definir operações específicas e um elemento identidade.

Há diversos sistemas de criptografia que utilizam chave pública, mas a maioria foi declarada comprovadamente insegura ou inviável em termos de uso prático. Atualmente, três tipos de sistemas de criptografias são considerados seguros com chaves públicas e eficientes: o primeiro é sistema de fatoração de inteiros (*Integer Factorization Systems* - IFS), o segundo é sistema de logaritmo discreto (*Discrete Logarithm System* - DLS) e o terceiro é sistema de curva elíptica (*Elliptic Curve Discrete Logarithm System* - ECDLS) (BARBOSA, 2003).

Tabela 3 - Exemplo de sistemas de chave pública (BARBOSA, 2003).

IFS	DLS	ECDLS
RSA Rabin-Williams	ElGamal DAS – US Gov. Diffie-Hellman/KE Schnorr	ECC

A Tabela 3 apresenta exemplo de alguns sistemas reais que utilizam as técnicas descritas anteriormente.

A Tabela 3 mostra os exemplos reais das técnicas e a Figura 4 apresenta o tempo necessário para violar um sistema ECC em comparação com as aplicações RSA e DSA. Esse esforço representado na Tabela 3 está apresentado em unidades de MIPS (um ano MIPS representa um ano do tempo da capacidade e desempenho de um milhão de instruções por segunda de uma de uma maquina) (BARBOSA, 2003).

Lembrando que uma medida considerada razoável como parâmetro de segurança é 10^{12} MIPS (10^{12} anos).

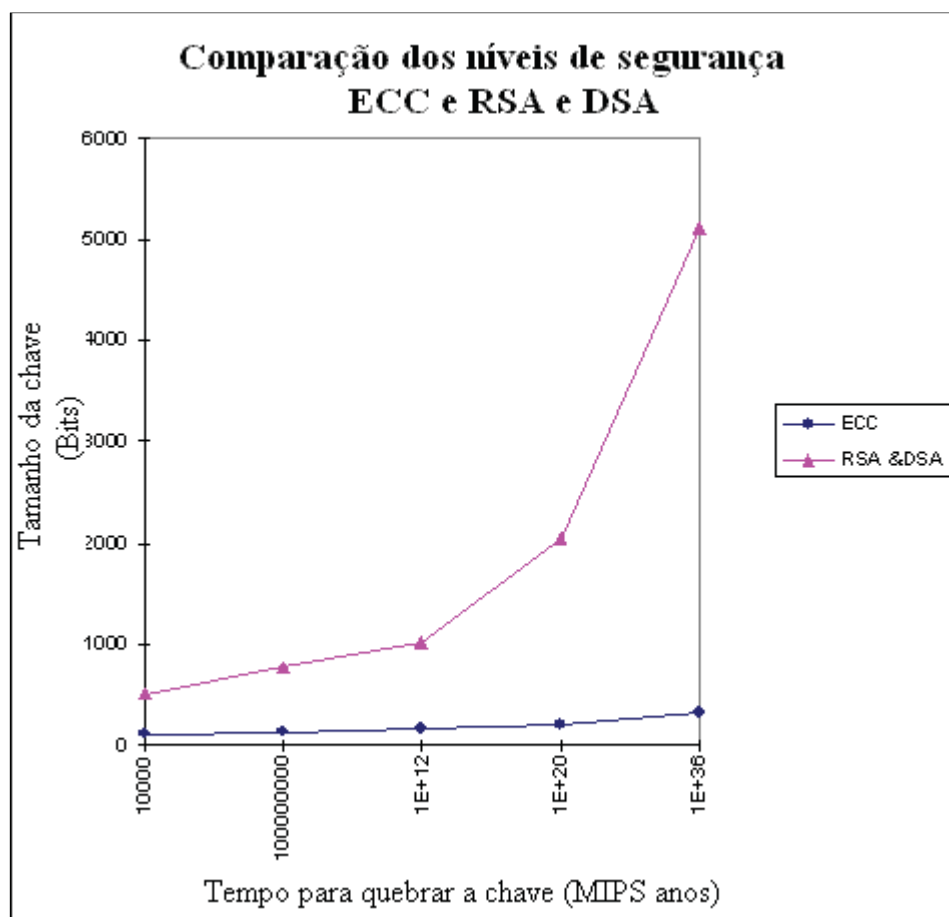


Figura 4 - Segurança X Tamanho de Chave (ECC, RSA e DSA)(BARBOSA, 2003)

São analisando a Figura 4 observa-se que, para um nível de razoável segurança (10^{12} MIPS), necessários 1024 bits para RSA e o DSA, enquanto que para ECC são necessários apenas 160 bits para o tamanho de chave. Mais um fato interessante é que conforme aumenta o nível de segurança (MIPS maior) o aumento do tamanho das chaves é proporcional para RSA e DSA, em comparação ao ECC. É fato que o aumento dos atuais parâmetros de segurança exigirá um crescimento considerável do tamanho da chave bem mais significativo para RSA e DSA do que no ECC.

3.3.1 Método proposto para o desenvolvimento da Curva Elíptica

A curva elíptica tem em sua forma geral:

$$y^2 = x^3 + ax + b$$

Onde x e y são elementos da curva e a e b são restrições para a mesma, sendo que a e b são inteiros módulo de H, que satisfazem:

$$4a^3 + 27b^2 \neq 0 \pmod{H}$$

Onde p é inteiro modulo principal da curva elíptica finita. Esta curva elíptica consiste em solucionar (x, y) definida pelas fórmulas apresentadas, juntamente com um elemento adicional chamado O , que é o ponto de curva elíptica no infinito. O conjunto de pontos (x, y) então assume coordenadas representadas na Tabela 5.

As operações básicas da curva elíptica são ponto adicional e ponto duplo. Pode-se dizer que para cada dado um ponto $P(x, y)$ em uma curva elíptica, é necessário calcular kP , onde k é um inteiro positivo. Para isso é necessária uma série de cálculos de multiplicação e adição de P . A variável $k=13$, determina a sequência necessária de operações, pelo qual P deve passar para chegar no ponto esperado.

Tabela 4 - Demonstrativo do Cálculo da adição e multiplicação.

P	2P	3P	6P	12P	13P
	Multiplica	Adiciona	Multiplica	Multiplica	Adiciona

O cálculo para realizar a multiplicação de dois pontos, sendo $P(x_p, y_p)$:

$$S = [(3x_p^2 + a)/2y_p] \pmod{H}$$

Então, $2P$ tem-se as coordenadas (x_r, y_r) :

$$x_r = (S^2 - 2x_p) \pmod{H}$$

$$y_r = [S(x_p - x_r) - y_p] \pmod{H}$$

E para determinar $3P$, é necessário adicionar ponto em $2P$, $2P = Q$, para isso passa-se $P(x_p, y_p)$ original e o valor vindo do $2P(x_q, y_q)$:

$$S = [(y_q - y_p) / (x_q - x_p)] \pmod{H}$$

$$x_r = (S^2 - x_p - x_q) \pmod{H}$$

$$y_r = (S(x_p - x_r) - y_p) \pmod{H}$$

Portanto, aplica-se a multiplicação ou a adição dependendo do valor de K . O ponto (x_r, y_r) é um ponto da curva elíptica. (VIGILA; MUNESWARAN, 2009)

3.3.2 Desenvolvimento do Cálculo da Curva Elíptica

A curva elíptica é representada pela fórmula:

$$y^2 \bmod H = (x^3 + x + 1) \bmod H$$

Esta curva retorna um conjunto de pontos que são mostrados na Tabela 5, considerando a e b com valor 1 e modulo 37 :

Tabela 5 - Dados retirados da fórmula pelos Scilab.

(0,1)	(0,36)	(21,25)	(21,12)
(1,15)	(1,22)	(24,14)	(24,23)
(2,14)	(2,23)	(25,0)	(25,0)
(6,36)	(6,1)	(26,18)	(26,19)
(8,15)	(8,22)	(27,8)	(27,29)
(9,31)	(9,6)	(28,15)	(28,22)
(10,7)	(10,30)	(29,31)	(29,6)
(11,14)	(11,23)	(30,24)	(30,13)
(13,18)	(13,19)	(31,36)	(31,1)
(14,24)	(14,13)	(33,9)	(33,28)
(17,11)	(17,26)	(35,18)	(35,19)
(19,16)	(19,21)	(36,6)	(36,31)

Dois pontos devem ser selecionados, um deles é G que é o ponto base, que neste caso foi utilizado o (0,1), este ponto implica que tem o menor (x,y) que satisfazem a curva elíptica. O segundo ponto é o PM, este ponto é aleatório que satisfaz a curva elíptica. Poderia escolher os dois pontos de mesmo valor, mas para fins de identificação individual os dois pontos têm valores diferentes. Neste caso o PM = (1,15). A escolha do PM é um exercício envolvendo aplicações meticolosas do processo das curvas elípticas.

O método da Curva Elíptica requer a seleção de um inteiro randômico k , onde $k < p$, e o mesmo k deve ser mantido em segredo. Devemos também selecionar a chave secreta chamada de n_b . Os valores k e n_b podem ser gerados por geradores de números aleatórios para dar credibilidade. Para esta análise será utilizado $k = 13$ e $n_b = 17$. A chave pública de B é validada:

$$P_b = n_b G$$

Supondo que se deseja criptografar uma mensagem de A para B, e A gostaria de enviar “w”, encontra-se o valor da tabela ASCII (codificação utilizada pelos computadores para representar os números e letras) referente a esta letra, que é 119, portanto:

$$P_b = n_b G = 17(0,1) = (21,12)$$

$$P_{ml} = 119(1,15) = (27,8)$$

As coordenadas de P_{ml} devem caber na curva elíptica. Esta transformação é feita para duas finalidades: primeira, o valor único da tabela ASCII é transformado em (x, y) das coordenadas na curva elíptica, e segunda é camuflar completamente os valores de possíveis pessoas mal intencionadas. A intenção é apresentar certo nível de complexidade, mesmo antes da mensagem ser criptografada de acordo com a curva elíptica.

O próximo passo é avaliar o kP_b , onde P_b é uma chave pública usada em B. Para determinar este produto foi utilizada uma série de multiplicação e soma, dependendo do valor de k . Sendo assim:

$$kP_b = 13(21,12) = (21,12)$$

$$P_{ml} + kP_b = (27,8) + (21,12) = (10,30)$$

$$KG = 13(0,1) = (0,1)$$

A mensagem criptografada é derivada da soma de P_{ml} com kP_b , isto é, $P_{ml} + kP_b$. Isso gera as coordenadas (x_2, y_2) . Então KG é incluído como o primeiro elemento (x_1, y_1) da versão criptografada, assim toda a versão criptografada para fins de armazenamento ou transmissão consistem de dois conjuntos de coordenadas é calculada da seguinte forma:

$$C_m = (KG, P_{ml} + kP_b)$$

$$KG = x_1, y_1$$

$$P_{ml} + kP_b = x_2, y_2$$

A mensagem criptografada é: $((x_1, y_1), (x_2, y_2)) = (0,1),(10,30)$, onde $x_1 = 0, y_1 = 1, x_2 = 10$ e $y_2 = 30$. (VIGILA; MUNEESWARAN, 2009)

3.4 Logaritmo Discreto

Com a dificuldade da descryptografia de alguns sistemas criptográficos, muitos se basearam no logaritmo discreto para sua descryptografia pela sua segurança em grupos de ordem primos. Como exemplos, o protocolo de Diffie e Hellman, o sistema de ElGamal, o DSA e as Curvas Elípticas.

Para exemplificar esta dificuldade apresentada é utilizada a técnica de logaritmo discreto nas curvas elípticas nos próximos tópicos para abordar seus problemas. A partir de agora será adotada a definição do problema do logaritmo discreto de Marques (2007).

(O problema do logaritmo discreto em curvas elípticas (PLDCE)) Dada uma curva elíptica E , definida sobre um corpo finito F_q , um ponto $P \in E(F_q)$ (de ordem n) e um ponto $Q \in E$, o problema do logaritmo discreto em curvas elípticas consiste em encontrar o inteiro $l \in [0, n - 1]$ tal que $Q = lP$. O inteiro l é chamado logaritmo discreto de Q na base P , sendo representado por $l = \log_P Q$. (MARQUES, 2007, p. 42)

A definição original dos problemas com logaritmo discreto se refere a grupos multiplicativos, diferentemente da abordagem feita por Marques (2007) que utiliza grupo aditivo, mas nos casos de grupos de pontos de uma curva as propriedades entre eles são equivalentes.

Apesar da dificuldade de se trabalhar com logaritmo discreto com as curvas elípticas pelos seus cálculos complexos, identificou-se alguns benefícios como a utilização de parâmetros criptográficos menores e ainda manter o mesmo nível de segurança, podendo assim ser utilizado em circunstâncias com poucos recursos, como no caso de *smart cards*.

A seguir, será descrito o desenvolvimento do logaritmo discreto utilizando o método de Pollard.

3.4.1 Método Pollard

O método ρ de Pollard é uma técnica randomizada que pode ser aplicada a um grupo cíclico qualquer conforme descrito por Marques (2007). Assim, seja o grupo G um conjunto que, juntamente com uma operação binária bem definida, forma um grupo cíclico de

ordem prima n . G é inicialmente dividido em três partições S_1 , S_2 e S_3 , de cardinalidades aproximadamente iguais. Uma condição para tal divisão é que 1 não pertença S_2 .

O método Pollard se diferencia dos demais por não demandar recursos de espaço para armazenamento de dados.

O passo seguinte consiste no cálculo da sequência $x_0, x_1, x_2, \dots$, composto por elementos de G , e com $x_0 = 1$ e

$$x_{i+1} = f(x_i) = \begin{cases} \beta \cdot x_i, & \text{se } x_i \in S_1, \\ x_i^2, & \text{se } x_i \in S_2, \\ \alpha \cdot x_i, & \text{se } x_i \in S_3, \end{cases}$$

$$a_{i+1} = \begin{cases} a_i, & \text{se } x_i \in S_1, \\ 2a_i \bmod n, & \text{se } x_i \in S_2, \\ a_i + 1 \bmod n, & \text{se } x_i \in S_3, \end{cases}$$

para isso $i \geq 0$. Esta sequência é na verdade duas outras sequências $a_0, a_1, a_2, \dots$, e $b_0, b_1, b_2, \dots$, que satisfazem a relação $x_i = \alpha^{a_i} \beta^{b_i}$, para $i \geq 0$, com $a_0 = b_0 = 0$,

e

$$b_{i+1} = \begin{cases} b_i + 1 \bmod n, & \text{se } x_i \in S_1, \\ 2b_i \bmod n, & \text{se } x_i \in S_2, \\ b_i, & \text{se } x_i \in S_3. \end{cases}$$

É necessário utilizar o algoritmo “*Floyd’s cycle-finding*”, para encontrar o par (x_i, x_{2i}) , para tal $(x_{i+1}, x_{2i+2}) = (f(x_i), f(f(x_{2i})))$. Desta forma é possível encontrar dois elementos do grupo tais que $x_i = x_{2i}$. Assim, $\alpha^{a_i} \beta^{b_i} = \alpha^{a_{2i}} \beta^{b_{2i}}$, o que implica em $\beta^{b_i - b_{2i}} = \alpha^{a_{2i} - a_i}$

$$\begin{aligned} \alpha^{a_i} \beta^{b_i} &= \alpha^{a_{2i}} \beta^{b_{2i}} \Rightarrow \beta^{b_i - b_{2i}} = \alpha^{a_{2i} - a_i} \\ &\Rightarrow \log_{\alpha} \beta^{b_i - b_{2i}} = \log_{\alpha} \alpha^{a_{2i} - a_i} \\ &\Rightarrow (b_i - b_{2i}) \log_{\alpha} \beta = a_{2i} - a_i \\ &\Rightarrow \log_{\alpha} \beta = \frac{a_{2i} - a_i}{b_i - b_{2i}}, \end{aligned}$$

devendo-se enfatizar que todas as operações ocorrem no módulo n . Desde que b_i diferente de $b_{2i} \pmod{n}$, então o logaritmo discreto estará calculado.

Deve-se considerar agora o problema no contexto de curvas elípticas. O algoritmo de Pollard deve encontrar os pares (c', d') e (c'', d'') tais que

$$c'P + d'Q = c''P + d''Q,$$

onde todas as operações ocorrem módulo n . Esta fórmula pode ser reescrita da seguinte forma:

$$(c' - c'')P = (d'' - d') = (d'' - d')lP,$$

de onde se conclui que:

$$(c' - c'') \equiv (d'' - d')l \pmod{n}.$$

Multiplicando ambos os lados da equivalência pelo inverso de $(d'' - d')$, obtém-se o logaritmo discreto $l = \log_p Q$:

$$l = (c' - c'') (d'' - d')^{-1} \pmod{n}$$

Primeiro deve-se escolher os pares (c', d') e (c'', d'') onde o c e d são escolhidos aleatoriamente no intervalo $[0, n - 1]$ e armazenados em c, d e $cP + dQ$ em uma lista, repetir o processo até obter $cP + dQ$ igual a alguma linha anterior.

Será descrito abaixo a estrutura do método Pollard para grupos de curvas elípticas (MARQUES, 2007).

Algoritmo:

- 1 - Seleciona um valor L
- 2 - Selecione uma função partição $H: \langle P \rangle \rightarrow \{1, 2, \dots, L\}$
- 3 - para j de 1 até L faça
- 4 - Selecione randomicamente $a_j, b_j \in [0, n-1]$
- 5 - Calcule $R_j = a_jP + b_jQ$
- 6 - fim para
- 7 - Selecione aleatoriamente c', d' de intervalo $[0, n-1]$ e calcule $X' = c'P + d'Q$
- 8 - $X'' \leftarrow X, c'' \leftarrow c', d'' \leftarrow d'$
- 9 - repita
- 10 - $j \leftarrow H(X')$
- 11 - $X' \leftarrow X' + R_j, c' \leftarrow c' + a_j \pmod{n}, d' \leftarrow d' + b_j \pmod{n}$

```

12 - para i de 1 até 2 faça
13 -    $j \leftarrow H(X'')$ 
14 -    $X'' \leftarrow X'' + R_j$ ,  $c'' \leftarrow c'' + a_j \bmod n$ ,  $d'' \leftarrow d'' + b_j \bmod n$ 
15 - fim para
16 - até  $X' = X''$ 
17 - se  $d' = d''$  então
18 -   retorne falha
19 - fim se
20 - retorne  $l = (c' - c'')(d'' - d')^{-1} \bmod n$ 

```

Quando se adapta o algoritmo de Pollard para solucionar esta situação, encontrado os pares (c', d') e (c'', d'') sem a necessidade de armazená-los, deve-se inicialmente estabelecer uma função f que seja aparentemente aleatória. Neste caso, $f: \langle P \rangle \rightarrow \langle P \rangle$, onde dados $X \in \langle P \rangle$ e $X = cP + dQ$, com $c, d \in [0, n - 1]$, f deve ser capaz de retornar iterativamente, $X' = c'P + d'Q$, com $c', d' \in [0, n - 1]$. A seguir, $\langle P \rangle$ é dividido em L conjuntos $\{S_1, S_2, \dots, S_L\}$, de cardinalidade aproximadamente iguais. Por fim, estabelece-se a função partição, dada por $H(X) = j$, que indica a partição que um dado elemento X pertence. Assim, a função $f(X)$ fica mais bem definida por

$$f(X) = X + a_j P + b_j Q,$$

onde $j = H(X)$ e a_j e b_j são randomicamente escolhidos em $[0, n-1]$.

Assim como no caso geral, as colisões são calculadas utilizando-se também do algoritmo “*Floyd’s cycle-finding*”, que determina o par (X_i, X_{2i}) até que seja encontrado um par em que $X_i = X_{2i}$. Sendo f uma função randômica, o número médio de testes executados para se encontrar uma colisão deste tipo é de $1,0308\sqrt{n}$ e o número de operações é de cerca de $3\sqrt{n}$. Todo o processo pode ser acompanhado no exemplo a seguir.

Exemplo: Seja a curva $E(F_{229}) : y^2 = x^3 + x + 44$. Como 229 é primo, o grupo de pontos desta curva é cíclico, tendo assim ordem $n = 229$. Seja $P = (2, 116) \in E(F_{229})$ e $Q = (155, 166) \in \langle P \rangle$. Neste exemplo são mostradas as etapas necessárias para o cálculo de $l = \text{Log}_P Q$, seguindo o algoritmo demonstrado acima.

Inicialmente, determina-se $L = 4$ e a função $H : \langle P \rangle \rightarrow \{1, 2, 3, 4\}$, dado por

$$H(x, y) = (x \bmod 4) + 1.$$

Executando o laço do exemplo, são gerados os seguintes valores:

$$[a_1, b_1, R_1] = [79, 163, (135, 117)]$$

$$[a_2, b_2, R_2] = [206, 19, (96, 97)]$$

$$[a_3, b_3, R_3] = [87, 109, (84, 62)]$$

$$[a_4, b_4, R_4] = [219, 69, (72, 134)]$$

Gerando os valores intermediários apresentados na Tabela 6. O valor adequado é encontrado na 12ª interação. Através desses valores, calcula-se:

$$l = (192 - 213) * (104 - 24)^{-1} \bmod 229 = 176,$$

Que corresponde ao valor do logaritmo discreto l .

Tabela 6 - Valores do passo intermediários do Algoritmo, aplicado ao exemplo.

i	c'	d'	X'	c''	d''	X''
-	54	175	(39,159)	54	175	(39,159)
1	34	4	(160, 9)	113	167	(130,182)
2	113	167	(130,182)	180	105	(36, 97)
3	200	37	(27, 17)	0	97	(108, 89)
4	180	105	(36, 97)	46	40	(223,15)
5	20	29	(119,180)	232	127	(167, 57)
6	0	97	(108, 89)	192	24	(57,105)
7	79	21	(81,168)	139	111	(185,227)
8	46	40	(223,153)	193	0	(197, 92)
9	26	108	(9, 18)	140	87	(194,145)
10	232	127	(167, 57)	67	120	(223,153)
11	212	195	(75,136)	14	207	(167, 57)
12	192	24	(57,105)	213	104	(57,105)

3.5 Considerações Finais

Neste capítulo foi apresentado o conceito de criptografia que pode-se definir sendo a ciência que utiliza números e expressões matemáticas para cifrar (encriptar) e decifrar (desencriptar) dados.

Outro ponto apresentado neste capítulo foi o desenvolvimento da ECC e sua descryptografia, onde foi demonstrado passo a passo todos os cálculos. Com isso o leitor está

apto a desenvolver tanto a criptografia como também a descriptografia das ECC seguindo o passo a passo.

No próximo capítulo vai ser apresentada a modelagem da proposta de melhoria na segurança nas redes VANETs, utilizando-se a modelagem UML e a modelagem em Fluxograma.

4. Proposta de um protocolo para comunicação em redes veiculares

4.1 Considerações iniciais

Considerando os estudos apresentados no capítulo 2 será proposta uma alteração na rede VANETs com objetivo de torná-la mais segura e dificultar os ataques por pessoas mal intencionadas.

Neste capítulo será apresentada a proposta de modelagem na rede VANET utilizando o protocolo TESLA e a criptografia ECC na forma de UML e fluxograma.

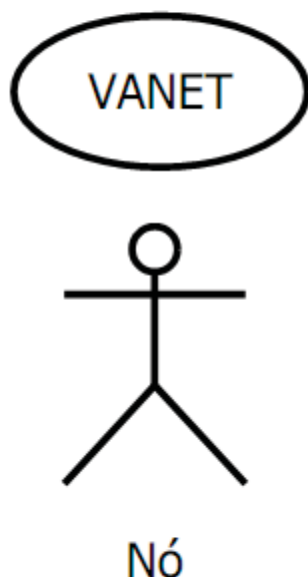
A seguir serão apresentados de uma forma simplificada as redes VANETs e os diagramas e as melhorias sugeridas.

4.2 Representação Simplificada

A seguir nas figuras 5, 6 e 7 serão apresentadas de forma simplificada a rede VANET, em seguida suas melhorias nos diagramas em UML.

VANET

VANET são redes Ad Hoc nas quais todos os veículos são conectados em uma rede para troca de dados.



Nó - São veículos ou pontos nas estradas, sendo que o mesmo se dá através de torres de comunicação de telefonia ou postes de acesso externo

Figura 5 - Representação da rede VANET

A Figura 5 mostra a rede VANET de uma forma simplificada, sendo que a mesma é demonstrada na forma de um objeto, este representa a rede que é composta por diversos nós e pontos na pista.

Esta representação simplificada da rede VANET, será dividida em blocos para um melhor entendimento do seu funcionamento.

Estrutura VANET

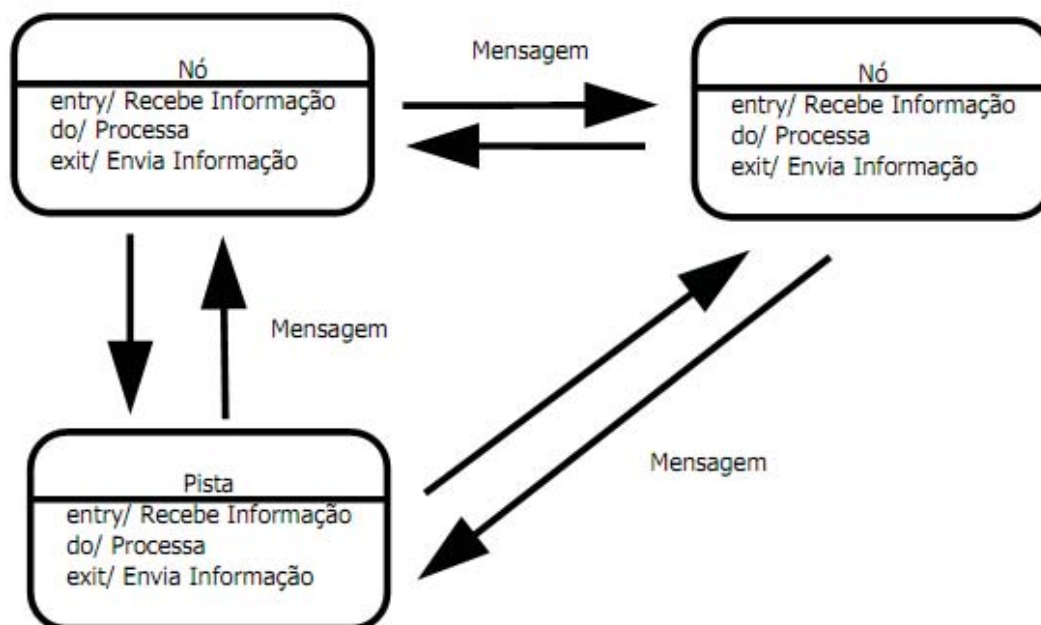
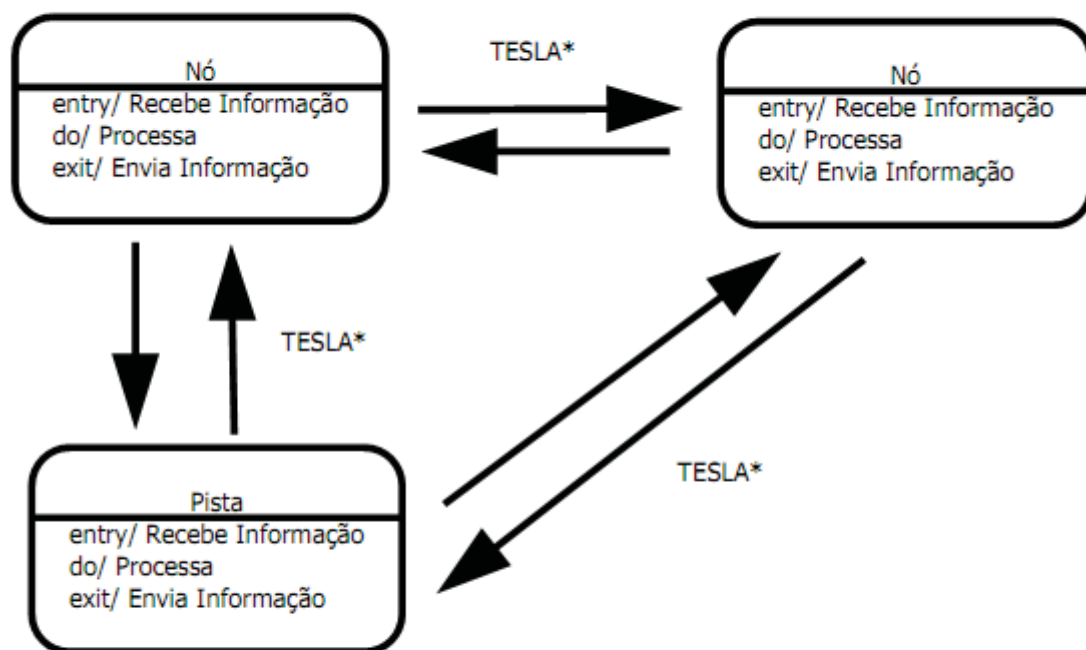


Figura 6 - Ilustração da estrutura da VANET.

Na Figura 6 é mostrada a primeira explosão do objeto VANET, ainda bem simplificado, mostrando apenas seu funcionamento na troca das informações. Observa-se que a estrutura da rede VANET demonstra que podem ser enviadas mensagens entre os nós e entre os nós e a pista, basicamente para todos os nós e pista ao seu redor.

Ainda na Figura 6 mostra que cada nó ou pista executam 3 ações de forma simplificada: I entry (ação de receber a informação de outros nós ou pista); II do (executa alguma ação caso necessária); III exit (ação de enviar alguma informação para outros nós ou pista).

VANET - TESLA



Para melhorar ainda mais a segurança já existente nas redes VANETs, este trabalho propõe a implementação do protocolo TESLA nas trocas de informações.

* TESLA é um protocolo de segurança na troca de informações na comunicação, baseado na perda de tempo de sincronização entre o emissor e o receptor.

Figura 7 - Demonstração da alteração da comunicação entre os Nós e Pista na rede VANET.

Na Figura 7 é mostrada a primeira explosão do objeto VANET como na Figura 6, mas com uma alteração na comunicação para melhorar a segurança entre os nós e nós e a pista, sendo esta alteração a substituição do protocolo a ser utilizado na rede VANET.

Observa-se que não houve alteração na estrutura conceitualmente, pois a rede VANET continua mandando informações entre os nós e nós e pista. A única alteração foi a forma de enviar esta informação.

4.3 Modelagem em UML

Para melhor entendimento das melhorias sugeridas neste trabalho será utilizada a modelagem UML (*Unified Modeling Language*), que é uma linguagem gráfica padrão para a

elaboração da estrutura de projetos complexos. Ela é utilizada para visualização, especificação, construção e elaboração de documentação.

A UML é o resultado da unificação de três linguagens de modelagem de objetos: *Booch*, *Object Modeling Technique* (OMT) e *Objected Oriented Software Engineering* (OOSE).

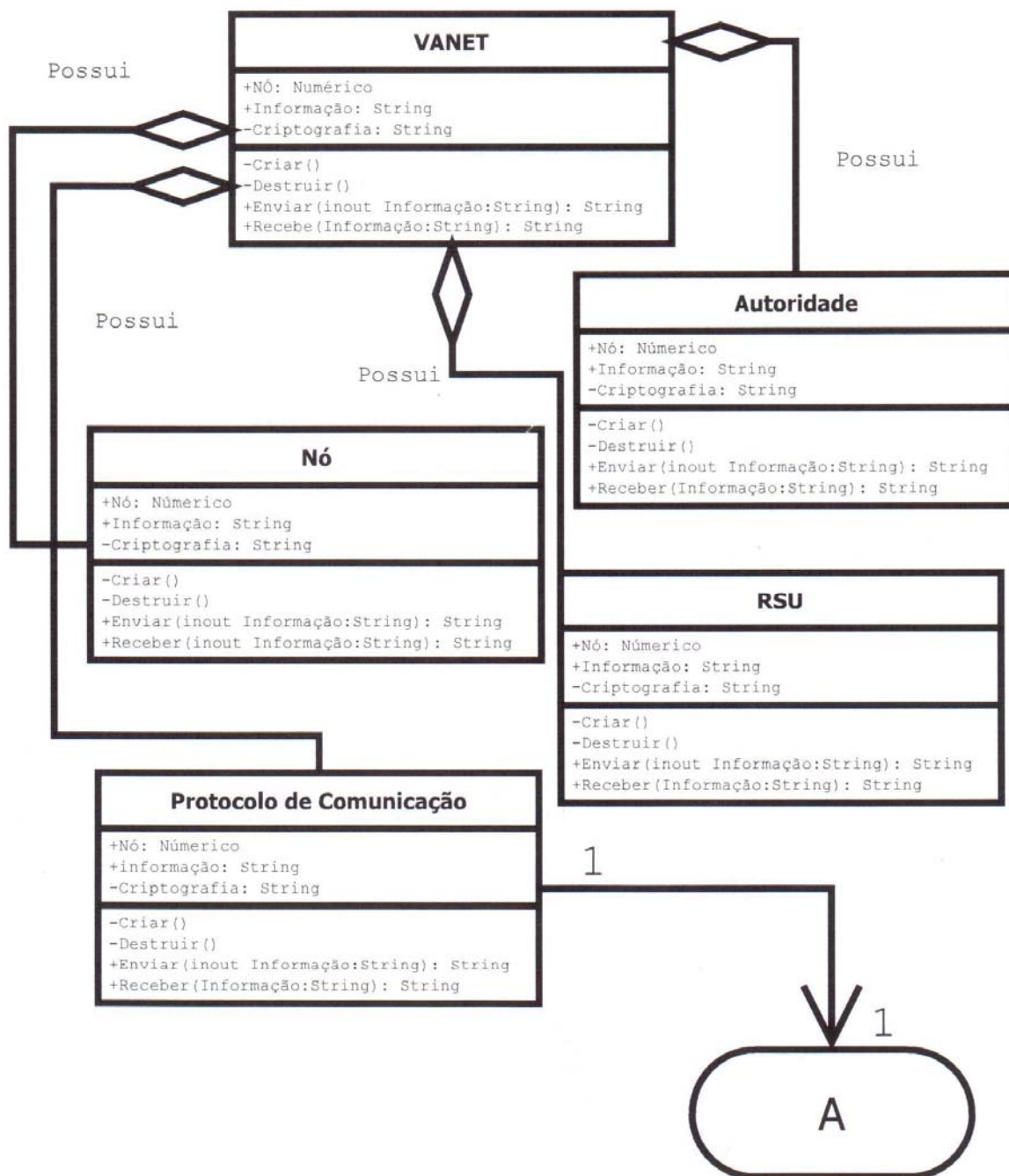
4.3.1 Diagrama de Classes

Nas Figuras 8 e 9 são apresentadas as classes alteradas para a proposta de melhoria, sendo elas VANET, TESLA e Criptografia. Inicialmente a classe VANET vai ser alterada para ter a possibilidade de escolher qual protocolo deseja ser utilizado. Em seguida a classe TESLA, que também será alterada para ter a possibilidade de escolher qual criptografia deseja utilizar.

A representação da letra A no diagrama mostra que continua o diagrama na próxima Figura a partir daquele ponto.

Diagrama de Classes

Melhoria - Incluir protocolo TESLA na rede VANET



Agregação, uma Classe faz parte de outra.

Associação, duas classes são associadas.

Figura 8 - Representação das classes utilizadas nas melhorias.

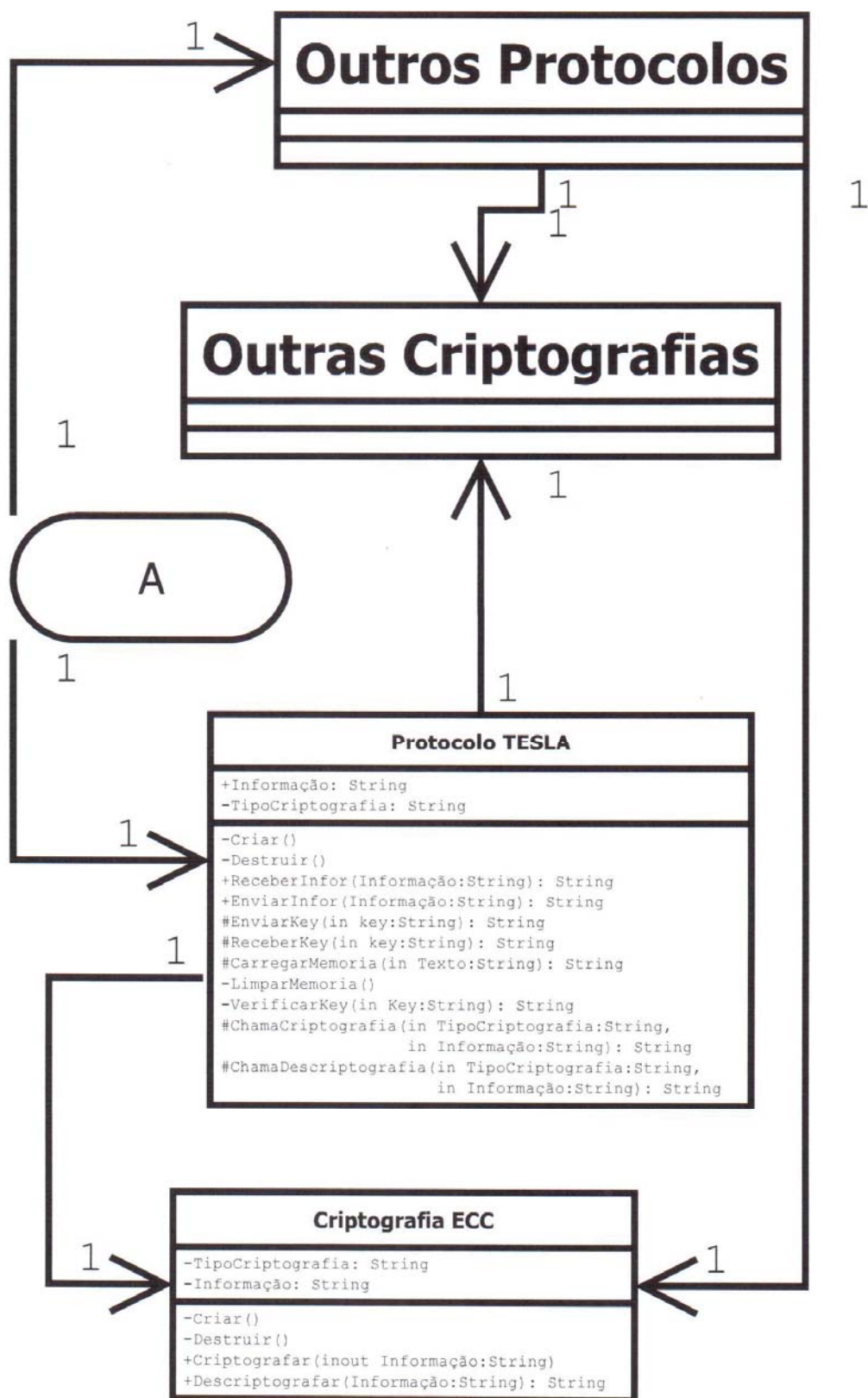


Figura 9 - Representação das classes utilizadas nas melhorias.

Nas Figuras 8 e 9 a apresentação da classe VANET e suas subdivisões ficam destacadas. Com isso pode-se observar as heranças nas classes Autoridade, Protocolo de comunicação, Nó e Pista. Tem-se na classe Protocolo de comunicação a ligação com o protocolo que vai ser utilizado, sendo este possível com qualquer protocolo ou com a classe Protocolo TESLA. Em seguida, é apresentada a ligação das criptografias, sendo que ambos os protocolos podem utilizar qualquer criptografia ou a criptografia ECC apresentada neste trabalho.

Estrutura Tesla com Envio

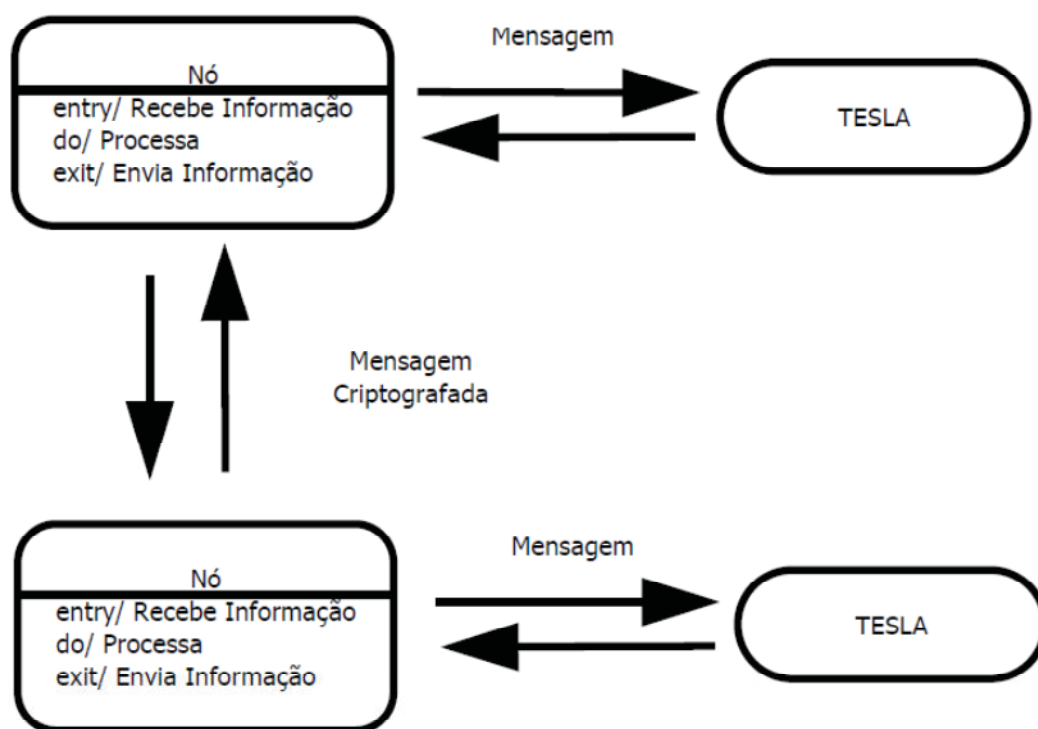


Figura 10 - Representação da Alteração do comportamento da Classe VANET.

A Figura 10 apresenta a forma que será enviada a mensagem. O primeiro passo é passar a mensagem para a classe TESLA que irá criptografar a mensagem e retornar para a classe nó ou pista para ser enviada para seu destinatário ou destinatários. Após o envio espera-se a chegada da mensagem no receptor para enviar a *key*, e em seguida o receptor chama a classe TESLA, e envia a mensagem criptografada e aguarda o envio da *key*, retornando em seguida a mensagem descriptografada.

4.3.2 Diagrama de Caso de Uso

Nas Figuras 11 e 12 são representados os métodos existentes nas classes TESLA e VANET:

Sendo eles: Receber Informações; Limpar Memória; Verificar *Key*; ChamarCriptografa; Carregar Memória; ChamarDescriptografa; Receber *Key*; Enviar *Key*; Enviar Informações;

Uma observação deste diagrama é que um relacionamento include de um caso de uso A para um caso de uso B indica que B é essencial para o comportamento de A.

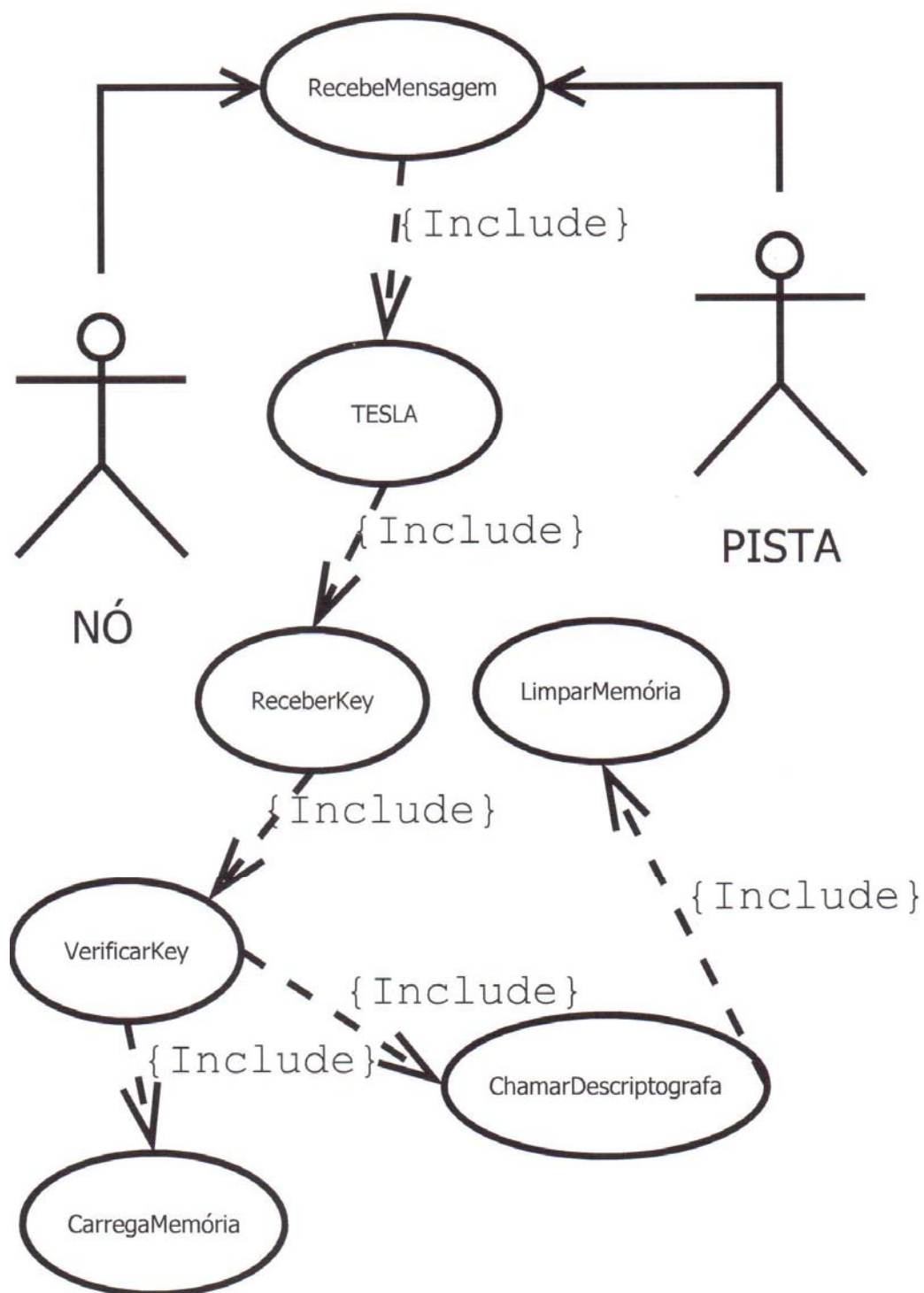


Figura 11 - Representação das funções recebe mensagem.

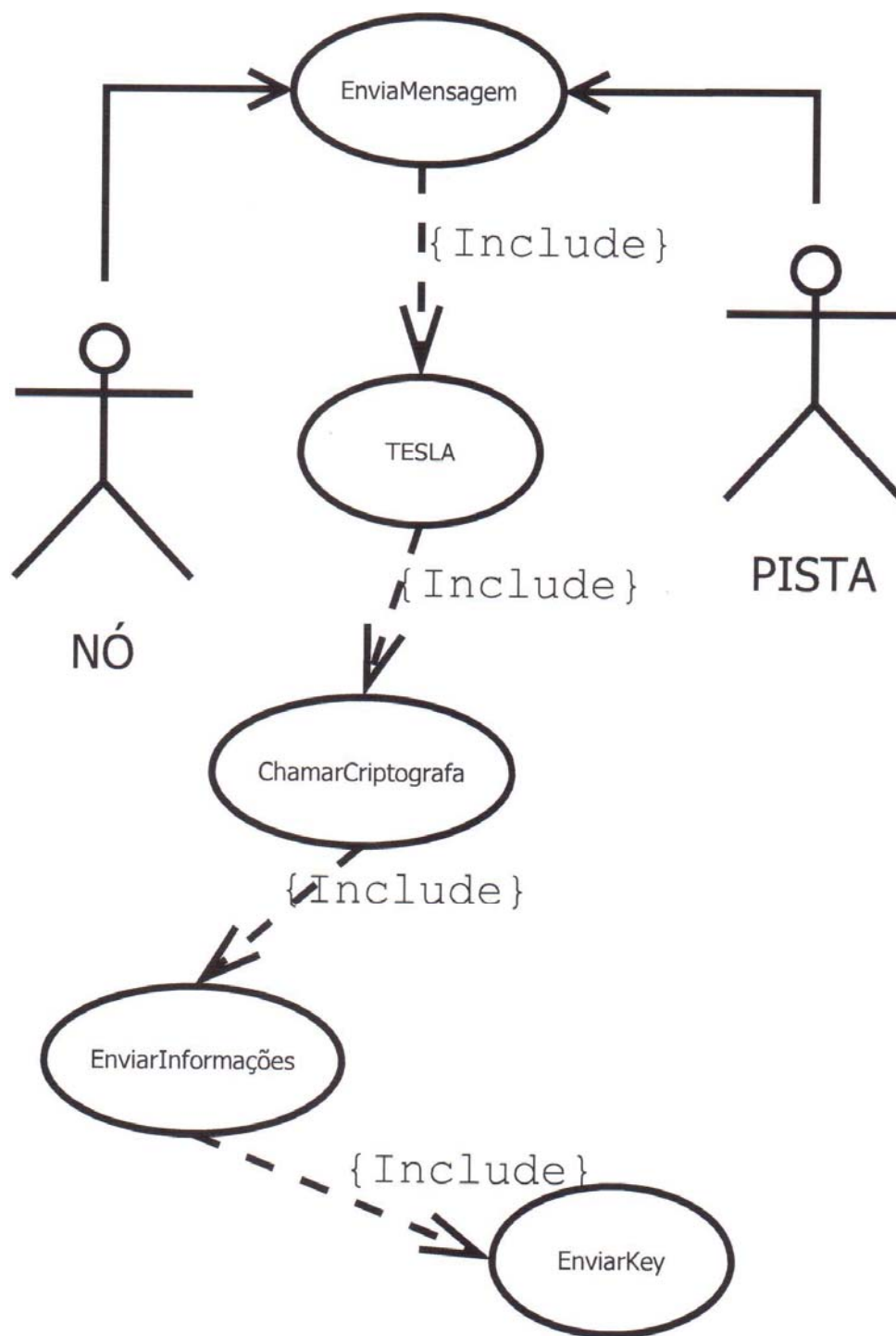


Figura 12 - Representação das funções envia mensagem.

4.3.3 Diagrama de Sequência

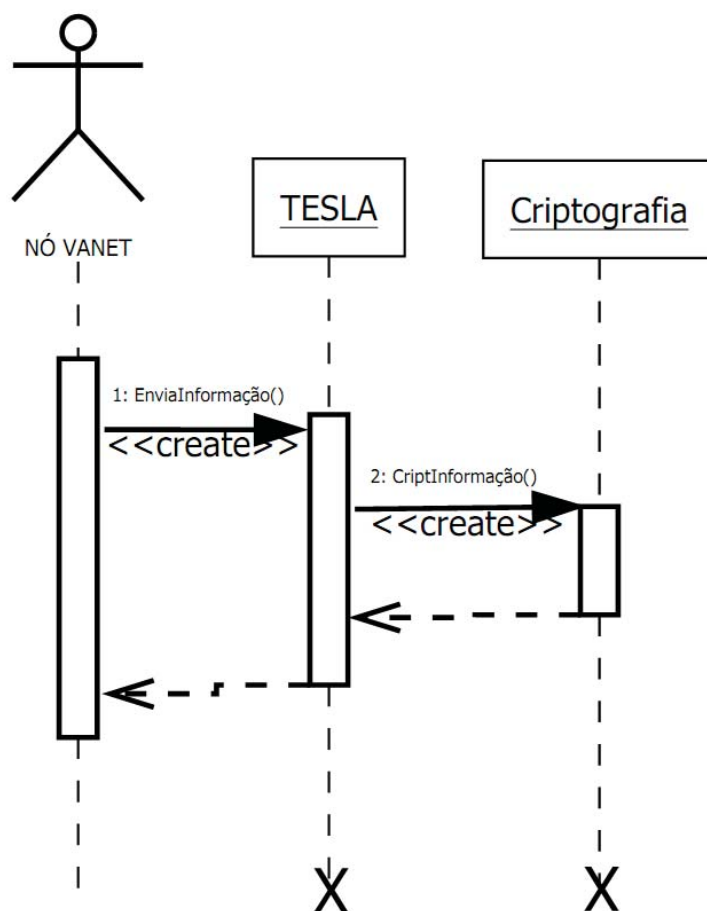


Figura 13 - Representação da sequência dos objetos.

A Figura 13 apresenta a sequência da utilização dos objetos pelo programa a ser desenvolvido.

A representação desta figura mostra o ciclo em que as classes são criadas e destruídas para não sobrecarregar a memória. Como é possível verificar nesta figura, a classe TESLA é removida da memória do emissor e receptor, pois após o envio da mensagem ou do recebimento da mesma não há necessidade de mantê-la. Juntamente com a classe TESLA a classe Criptografia é removida da memória. A única classe que não pode ser removida da memória é a classe VANET.

4.3.4 Diagrama de Estado

As Figuras 14 a 23 apresentam o fluxo de processo entre as classes VANET, TESLA e Criptografia, que são detalhadas para melhor entendimento da forma que deverá ser modelada nas redes VANETs. No apêndice A, pode-se visualizar o diagrama de estado completo. As letras no diagrama mostram que nas próximas Figuras aquele diagrama continua.

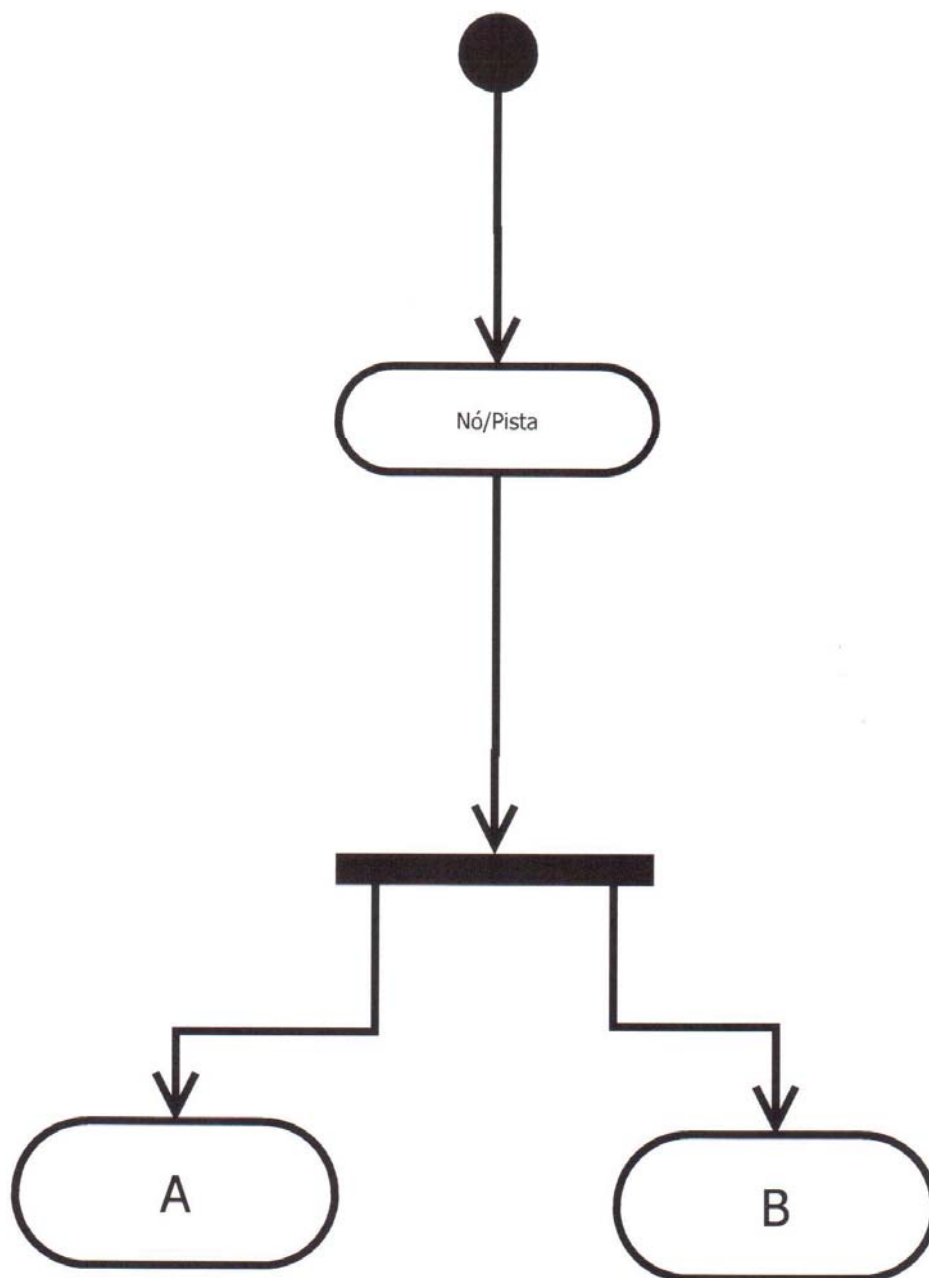


Figura 14 - Representação do diagrama de estado - Parte 1

A Figura 14 apresenta o início de um nó na rede VANET, que desencadeia dois processos paralelos, um deles é receber e o outro enviar mensagens.

Depois de iniciado um nó na rede VANET, é desencadeado um ciclo contínuo de receber e enviar mensagem, que só poderá ser interrompido se retirado a força.

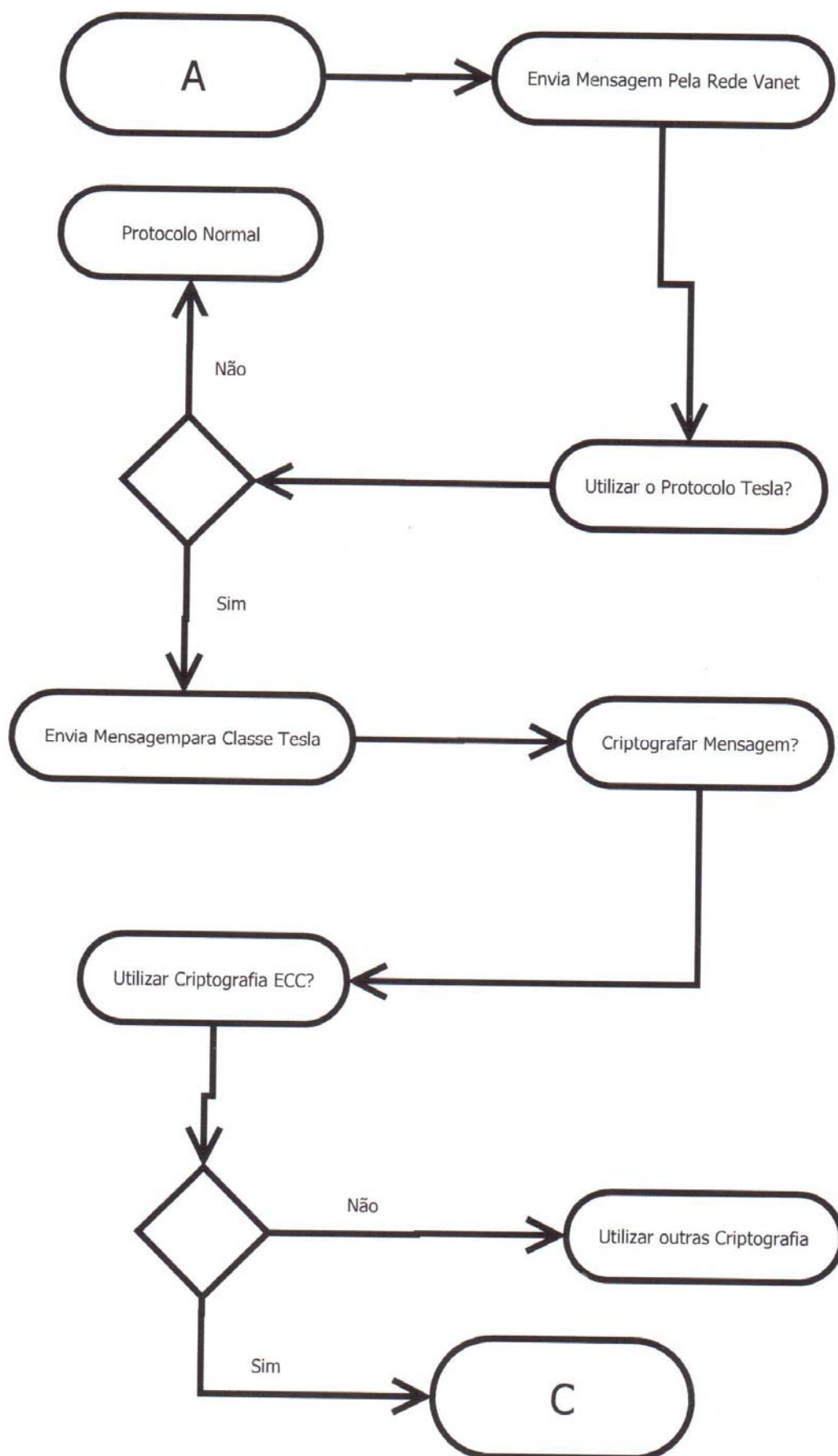


Figura 15 - Representação do diagrama de estado - Parte 2

A Figura 15 apresenta um dos processos paralelos que são executados no nó/pista no momento de envio de uma informação. A figura exemplifica que ao ter informação para ser enviada é questionado se vai ou não utilizar o protocolo TESLA, em caso positivo será direcionado para a melhoria, em caso negativo é utilizado um protocolo qualquer. Como o foco deste estudo é a melhoria na rede VANET com a utilização do protocolo TESLA, serão desconsiderados os outros protocolos e outras criptografias que não seja a ECC.

No caso de optar pelo protocolo TESLA, são enviadas as informações para a classe TESLA, em seguida será verificado se vai ser utilizada a criptografia de ECC, também sugerida neste trabalho como melhoria. Uma observação é que a classe ECC é independente do protocolo TESLA, podendo ser utilizado tanto com o protocolo TESLA quanto com qualquer outro protocolo, não sendo necessária a utilização apenas com ambas ao mesmo tempo.

Para um melhor entendimento deve-se considerar que ambas as alterações sugeridas foram escolhidas, com isso, a informação vai ser enviada para classe TESLA onde também será escolhida a criptografia ECC.

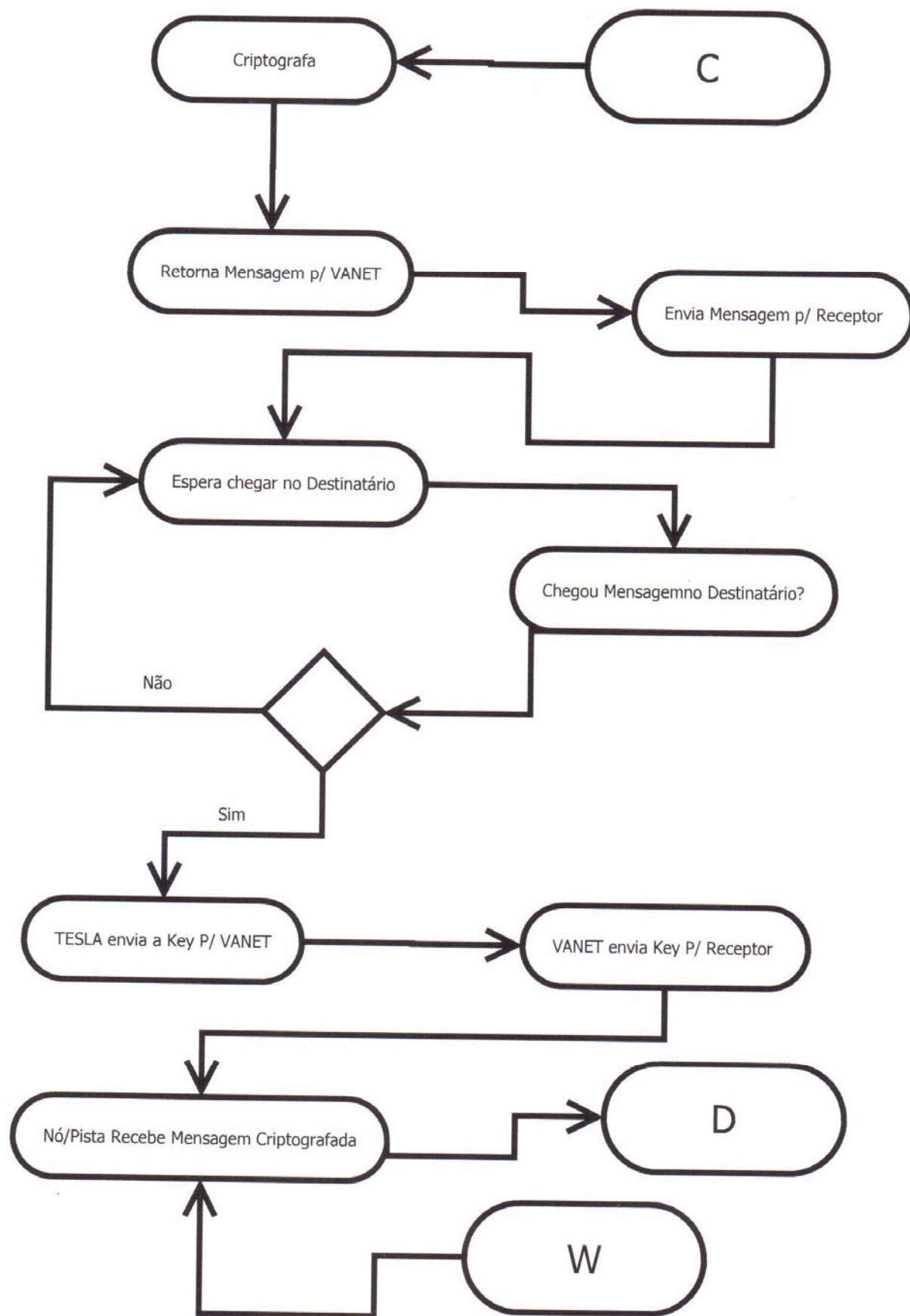


Figura 16 - Representação do diagrama de estado - Parte 3

A Figura 16 é a continuação da Figura 15 e mostra que após a escolha da criptografia a ser utilizada é enviada a informação para a classe Criptografia e logo após para a classe ECC.

Em seguida, a informação é retornada para a classe TESLA que retorna para a VANET, onde se espera o tempo da chegada da informação no destinatário, este tempo de espera é um cálculo feito pela classe TESLA, logo após é enviada a chave necessária para descriptografia da mensagem.

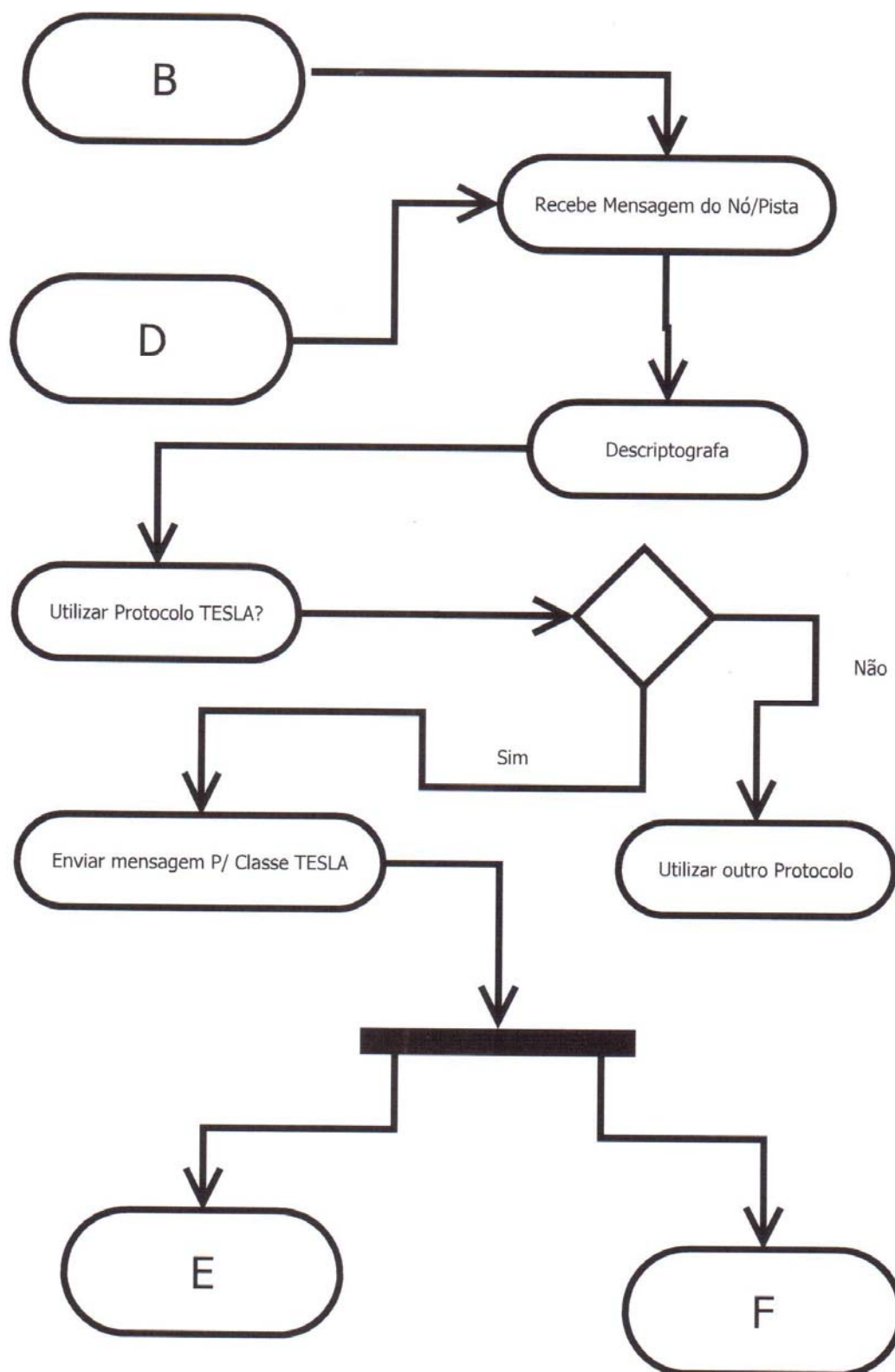


Figura 17 - Representação do diagrama de estado - Parte 4

A Figura 17 apresenta o segundo processo executado em paralelo com o processo de enviar dados para a rede. Nesta etapa a informação é recebida e da mesma forma que o envio verifica se está utilizando a classe TESLA e a criptografia ECC.

Para melhor entendimento considera-se que ambos estão sendo utilizados, e assim a informação chega a classe VANET, logo após é enviada para a classe TESLA que em seguida executa dois processos em paralelos descritos nas próximas figuras.

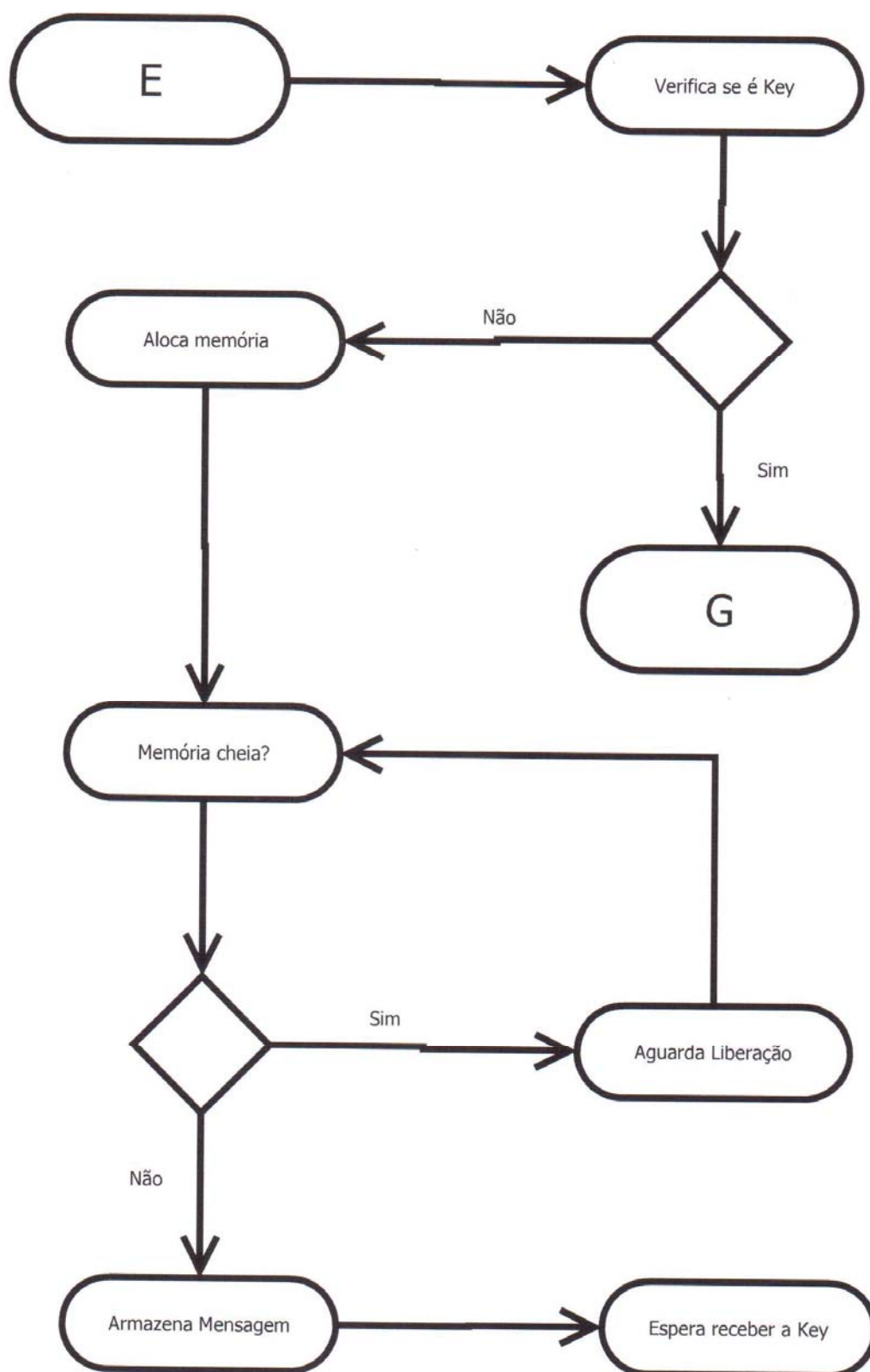


Figura 18 - Representação do diagrama de estado - Parte 5

Na Figura 18 é apresentado o primeiro processo executado em paralelo na classe TESLA. Neste processo a primeira etapa é verificar se a informação que a classe TESLA recebeu é a chave para descriptografar alguma mensagem que esta armazenada na memória ou é uma nova mensagem que deve ir para a memória para aguardar a chave.

Se não for a chave é verificado se tem memória suficiente para alocar, pois na utilização do protocolo TESLA a mensagem é alocada na memória por inteiro até a chegada da chave, se não tiver espaço suficiente na memória é necessário a espera até conseguir alocar o espaço suficiente para a nova mensagem, após alocar é armazenada a mensagem na memória até a chegada da chave.

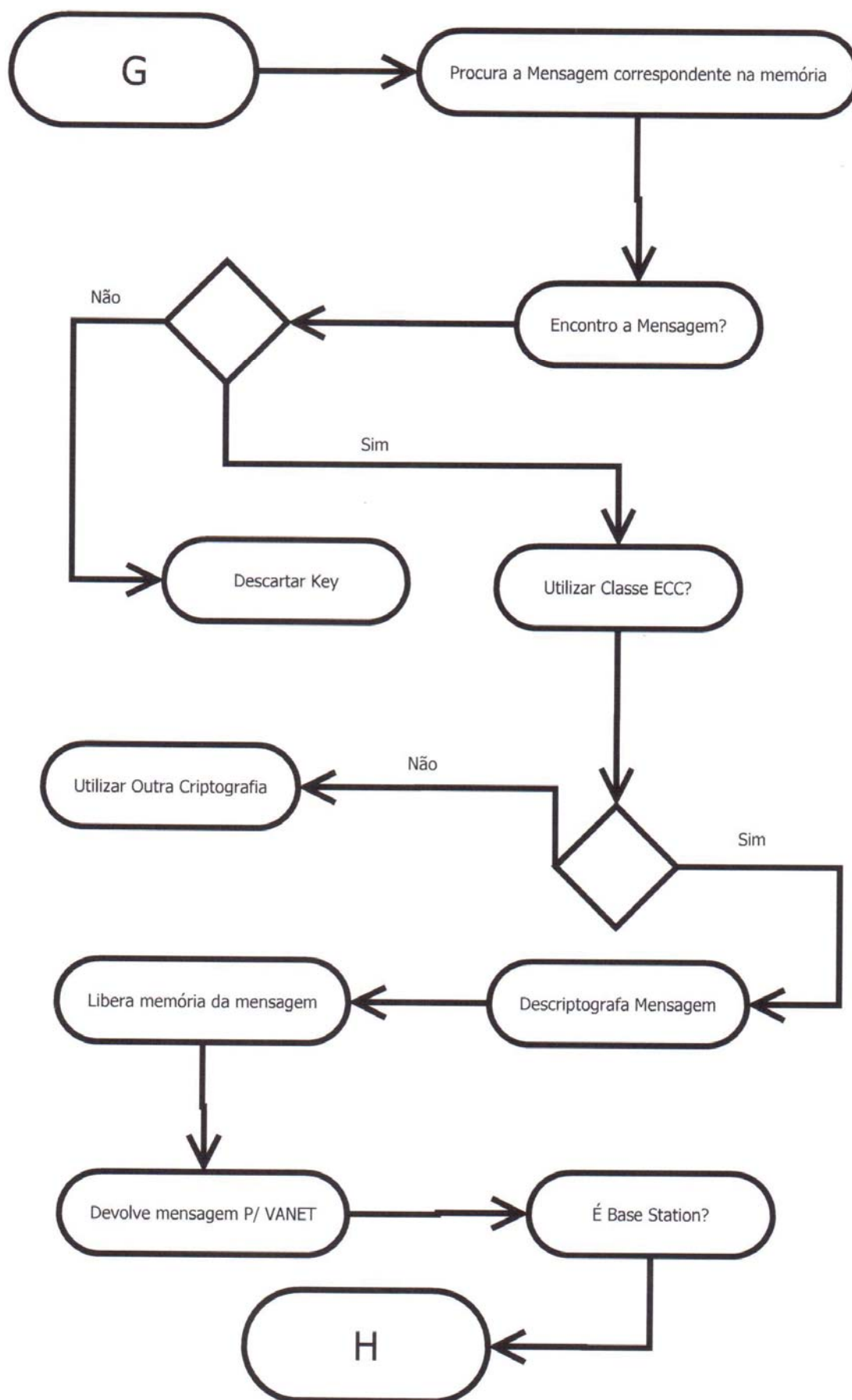


Figura 19 - Representação do diagrama de estado - Parte 6

A Figura 19 apresenta ainda o primeiro processo executado na classe TESLA, mas caso seja o recebimento da chave.

Primeiramente, procura na memória a mensagem que corresponde a chave que acabou de chegar, se não encontrar mensagem correspondente descarta a chave, senão verifica se está utilizando a classe da ECC para descriptografar a mensagem. A após libera a memória utilizada pela mensagem e devolve a mensagem descriptografada para a classe TESLA e logo após ocorre o retorno da mensagem para a classe VANET, onde se verifica se esta em um nó ou pista para ocorrer o tratamento necessário, se deve ser reenviada a mensagem ou armazenada.

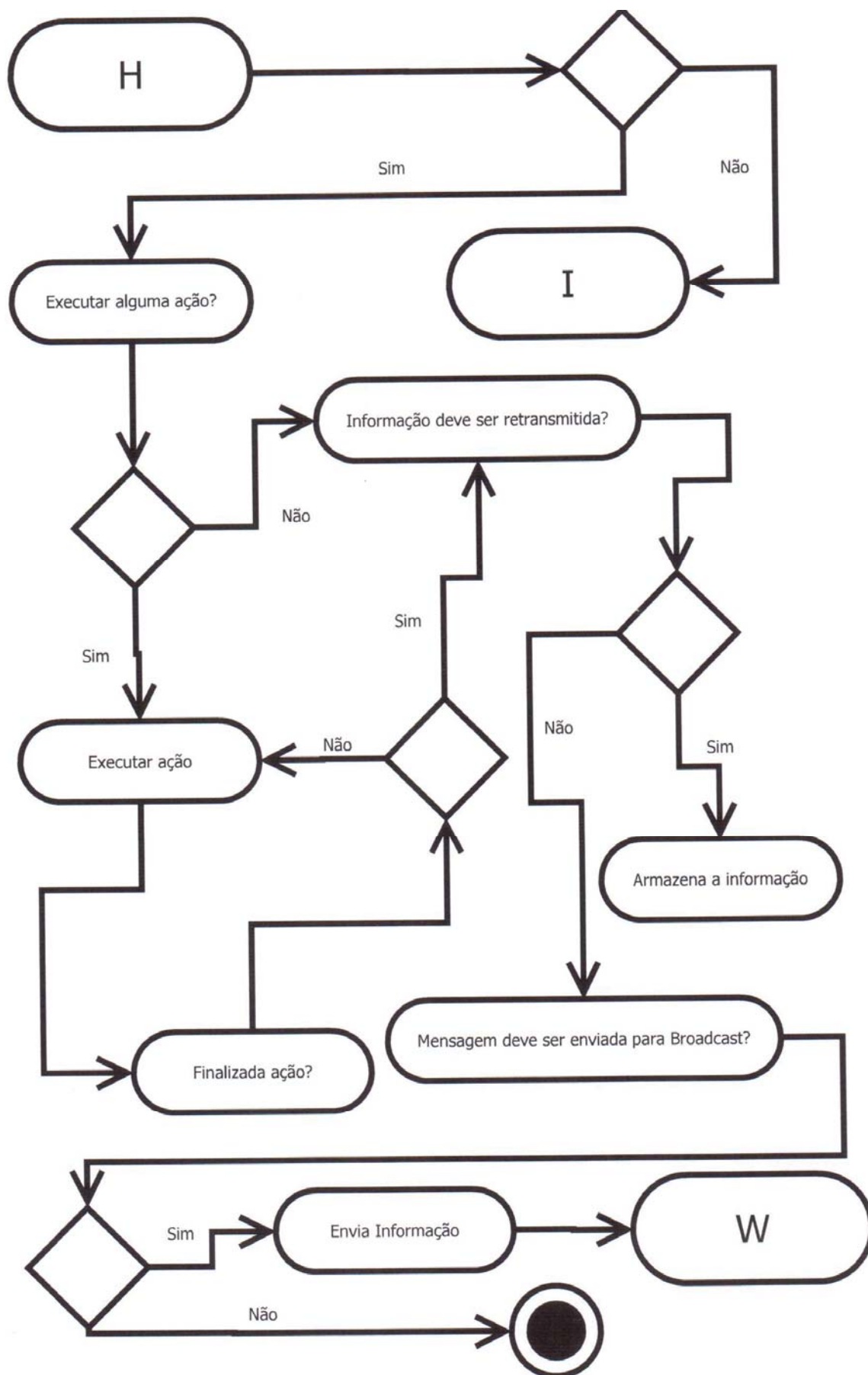


Figura 20 - Representação do diagrama de estado - Parte 7

A Figura 20 apresenta se a mensagem chegou na *base station*, então verifica se existe alguma ação que deve ser tomada, se sim executa a ação. A após, verifica se deve ser retransmitida a informação, se não armazena a informação enviada, se sim verifica se a mensagem deve ser enviada para *broadcast*.

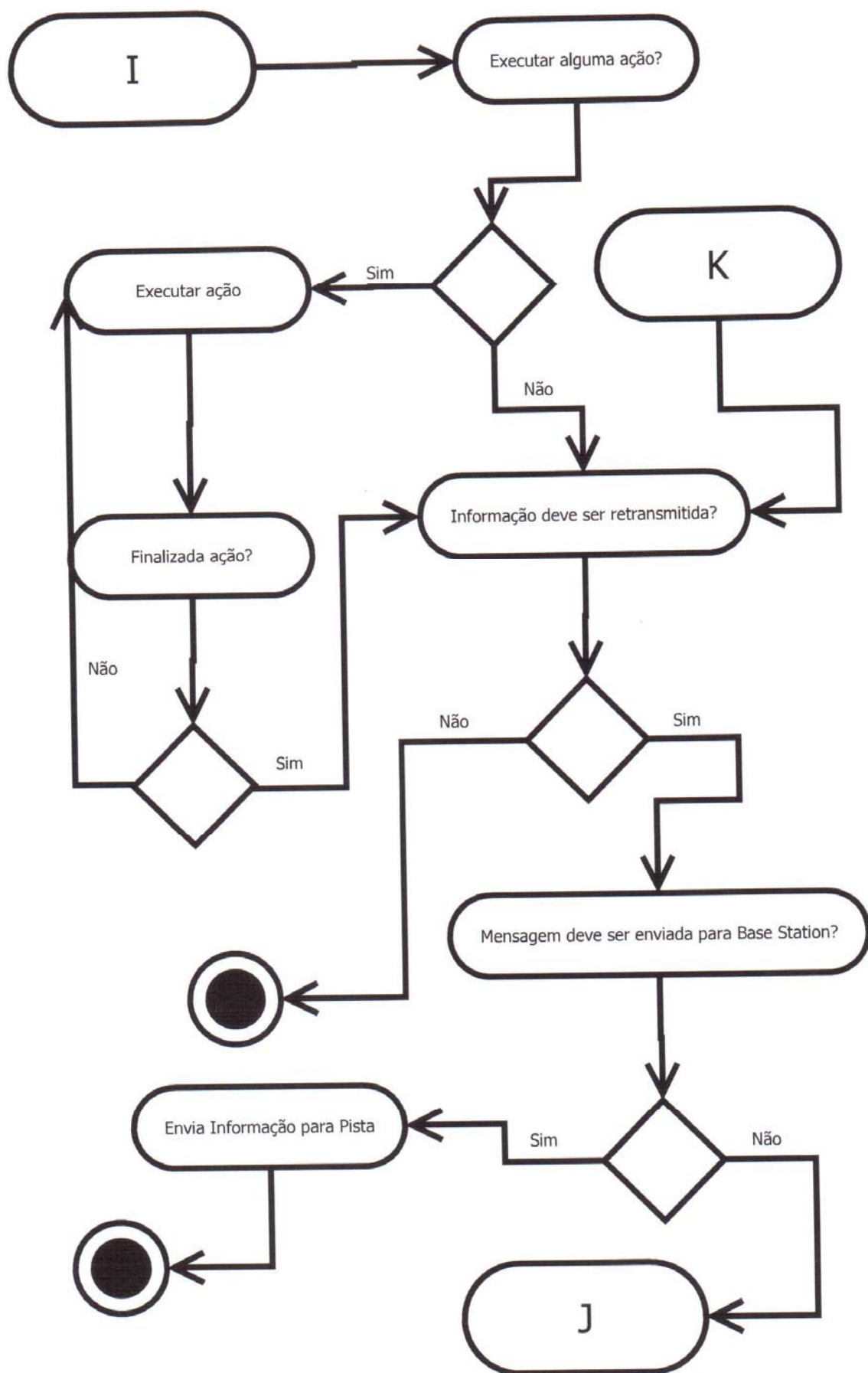


Figura 21 - Representação do diagrama de estado - Parte 8

A Figura 21 apresenta se a mensagem chegou na *broadcast*, verifica se existe alguma ação que deve ser tomada, se sim executa a ação. A após, verifica se deve ser retransmitida a informação, se não finaliza o processo desta informação enviada, se sim verifica se a mensagem deve ser enviada para *base station*, se sim envia apenas para a Pista, se não verifica se deve ser enviada também para *broadcast*.

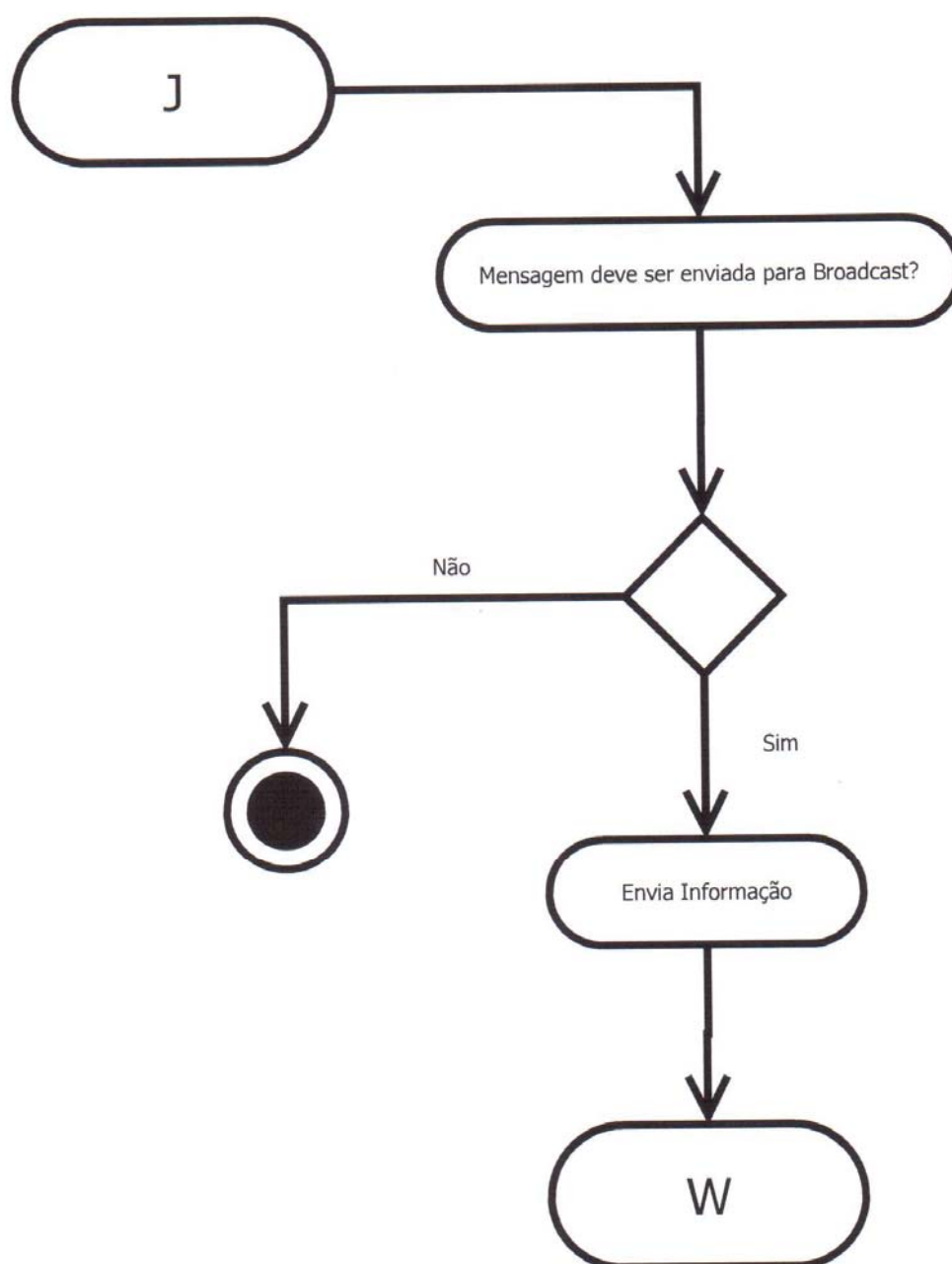


Figura 22 - Representação do diagrama de estado - Parte 9

A Figura 22 apresenta se a mensagem deve ser enviada para *broadcast* também após ser enviada para *base station*, se sim vai para um ciclo de envio.

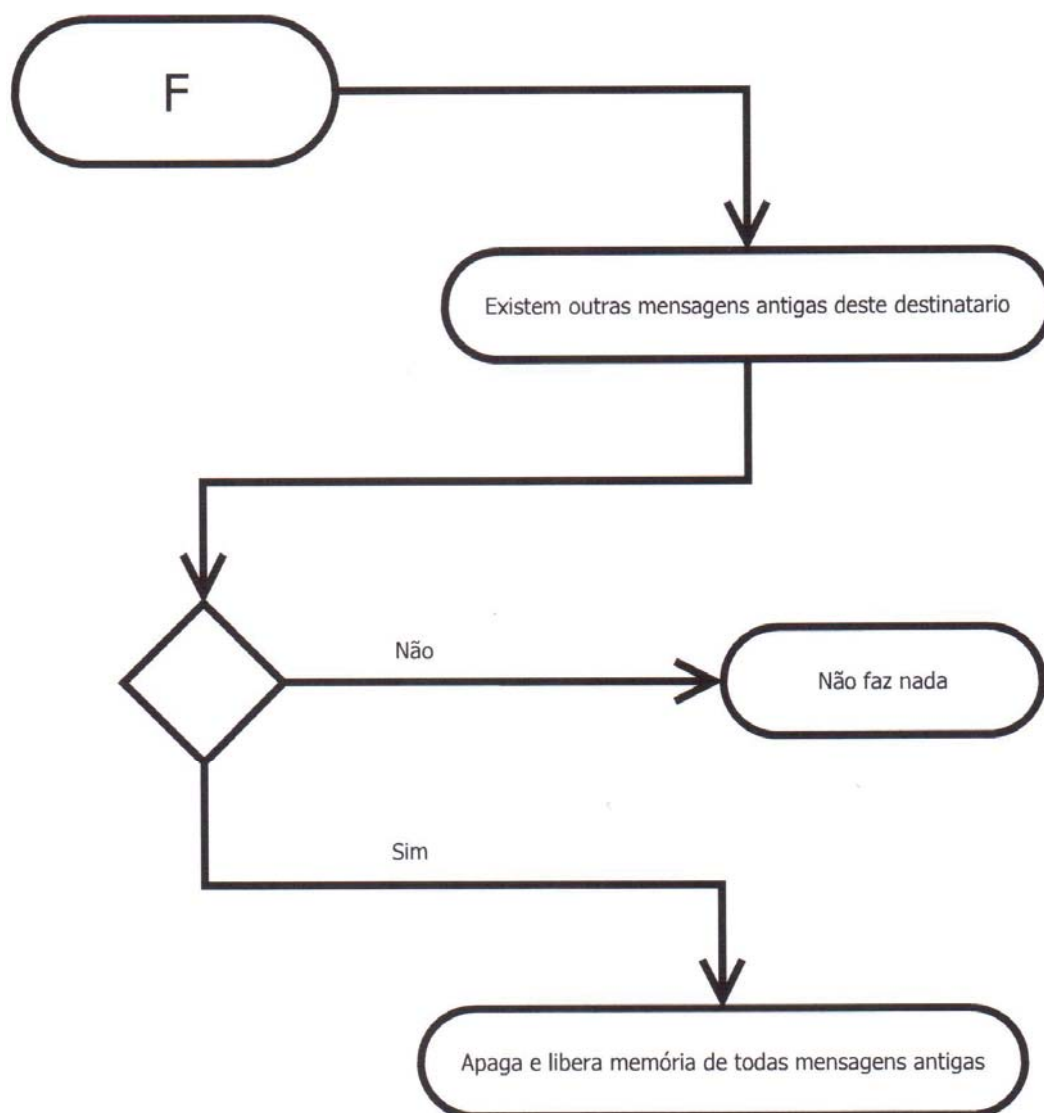


Figura 23 - Representação do diagrama de estado - Parte 10

Na Figura 23 é apresentado o segundo processo executado em paralelo na classe TESLA.

No momento que chega uma chave ou mensagem de um destinatário é feita a verificação para saber se não existem mensagens antigas do mesmo, pois se existir ocorreu a perda da chave e as mensagens devem ser desconsideradas e a memória liberada, por que não ocorrerá a chegada da mesma.

4.4 Fluxograma

Nesta seção é apresentada na forma de fluxograma a proposta de melhoria nas redes VANETs. O fluxograma é uma representação gráfica de um determinado processo ou

fluxo de trabalho, efetuado geralmente com recurso de figuras geométricas normalizadas e as setas unindo essas figuras geométricas. Através desta representação gráfica é possível compreender de forma rápida e fácil a transição de informações ou documentos entre os elementos que participam no processo em causa.

O fluxograma pode ser definido também como o gráfico em que se representa o percurso ou caminho percorrido por certo elemento (por exemplo, um determinado documento), através dos vários departamentos da organização, bem como o tratamento que cada um vai lhe dando.

A seguir será apresentado o fluxograma da melhoria subdividido em cinco partes. Estes representam o mesmo processo descrito na subsecção 4.2.4 e no Apêndice A.

- **O Fluxograma das redes VANETs**, apresentado na Figura 24, descreve o processo de forma macro, não são descritos os detalhes do protocolo TESLA, mas o processo por inteiro. Os detalhes de cada explosão deste fluxo são descrito nas Figuras 25, 26, 27 e 28.
- **O Fluxograma TESLA envia Mensagem**, apresentado na Figura 25, descreve com detalhes o processo do protocolo TESLA no momento de enviar uma mensagem criptografada a um receptor, tanto no momento da escolha da criptografia que vai ser utilizada na criptografia quanto ao momento de enviar a key para a descriptografar a mensagem no receptor.
- **Fluxograma TESLA recebe Mensagem**, apresentado na Figura 26, descreve os detalhes no momento que protocolo TESLA recebe uma mensagem que deve ser descriptografada, sendo que o protocolo controla o armazenamento da mensagem na memória no caso da mensagem, e a descriptografia da mensagem caso seja o key e logo após a liberação da memória em caso positivo do processo.
- **Fluxograma das Mensagens Recebidas no Nó**, apresentado na Figura 27, descreve os detalhes quando uma mensagem chega ao nó, sendo neste caso a verificação se vai ser reenviada a mensagem, em caso positivo deve ser reenviada para base station e/ou para broadcast.
- **Fluxograma das Mensagens Recebidas na Pista**, apresentado na Figura 28, descreve os detalhes quando uma mensagem chega em algum ponto da pista, sendo neste caso a verificação se alguma das informações devem ser armazenada, ou se devem ser reenviada para broadcast.

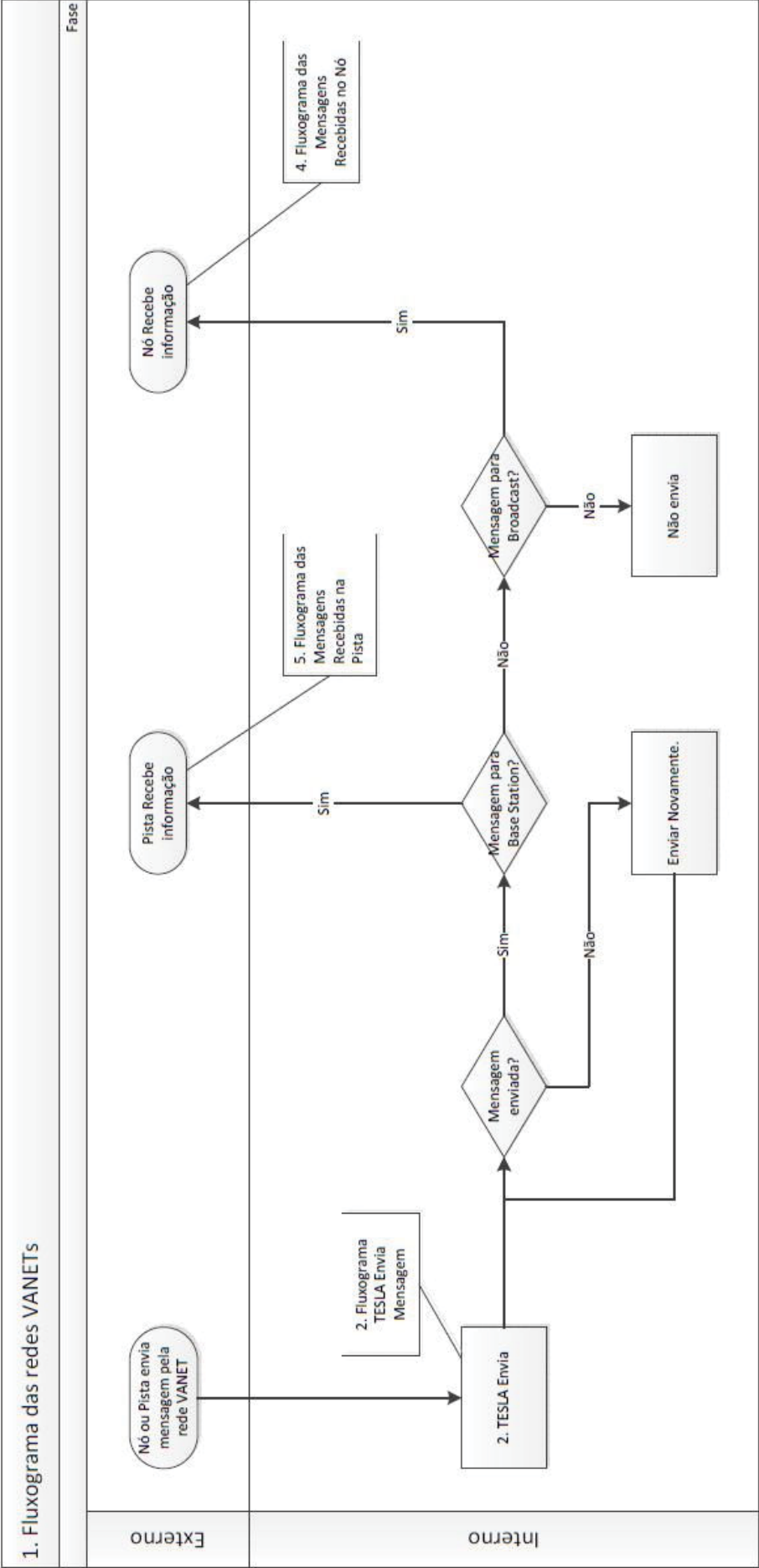


Figura 24 - Fluxograma das redes VANETs

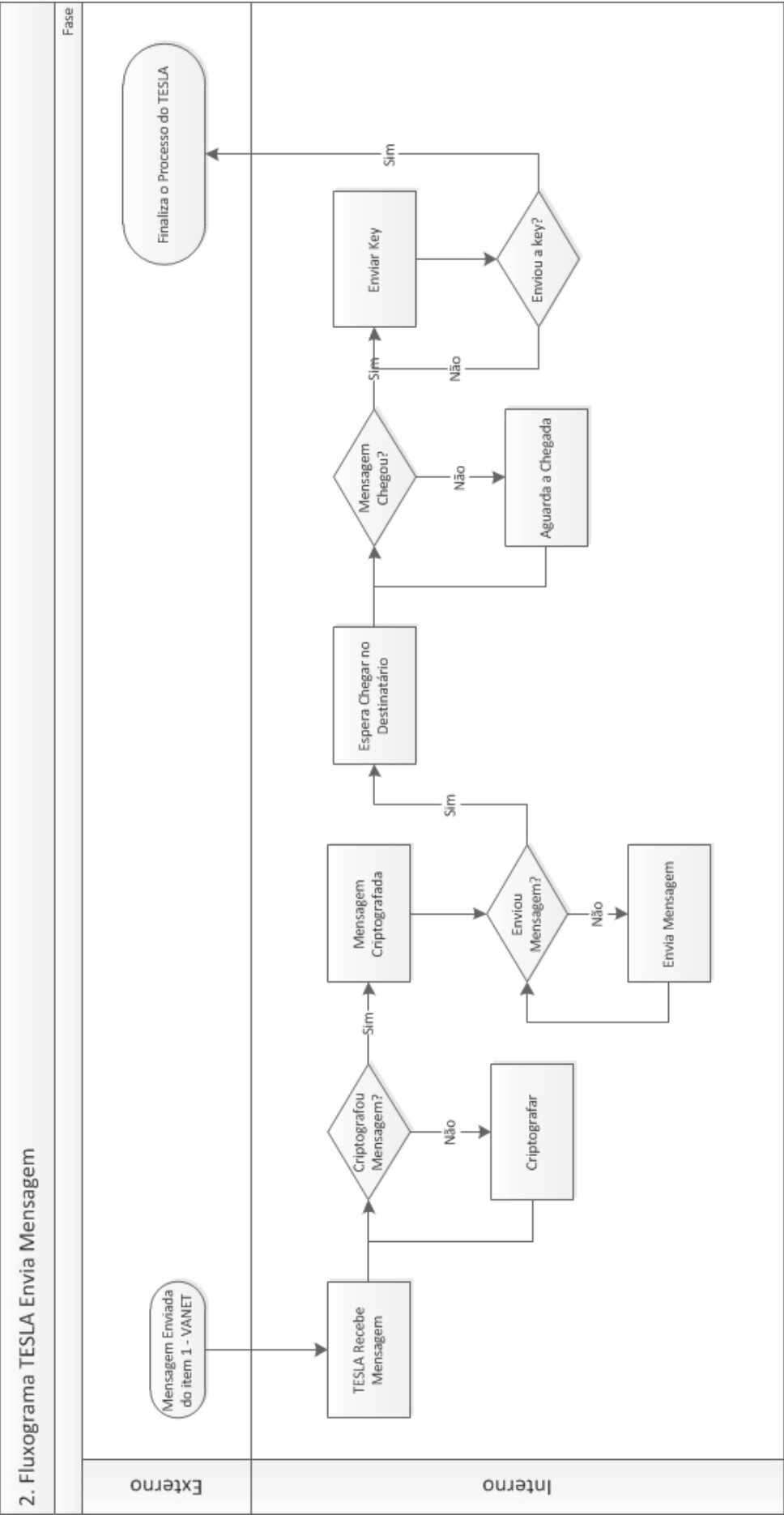


Figura 25 - Fluxograma TESLA Envia Mensagem

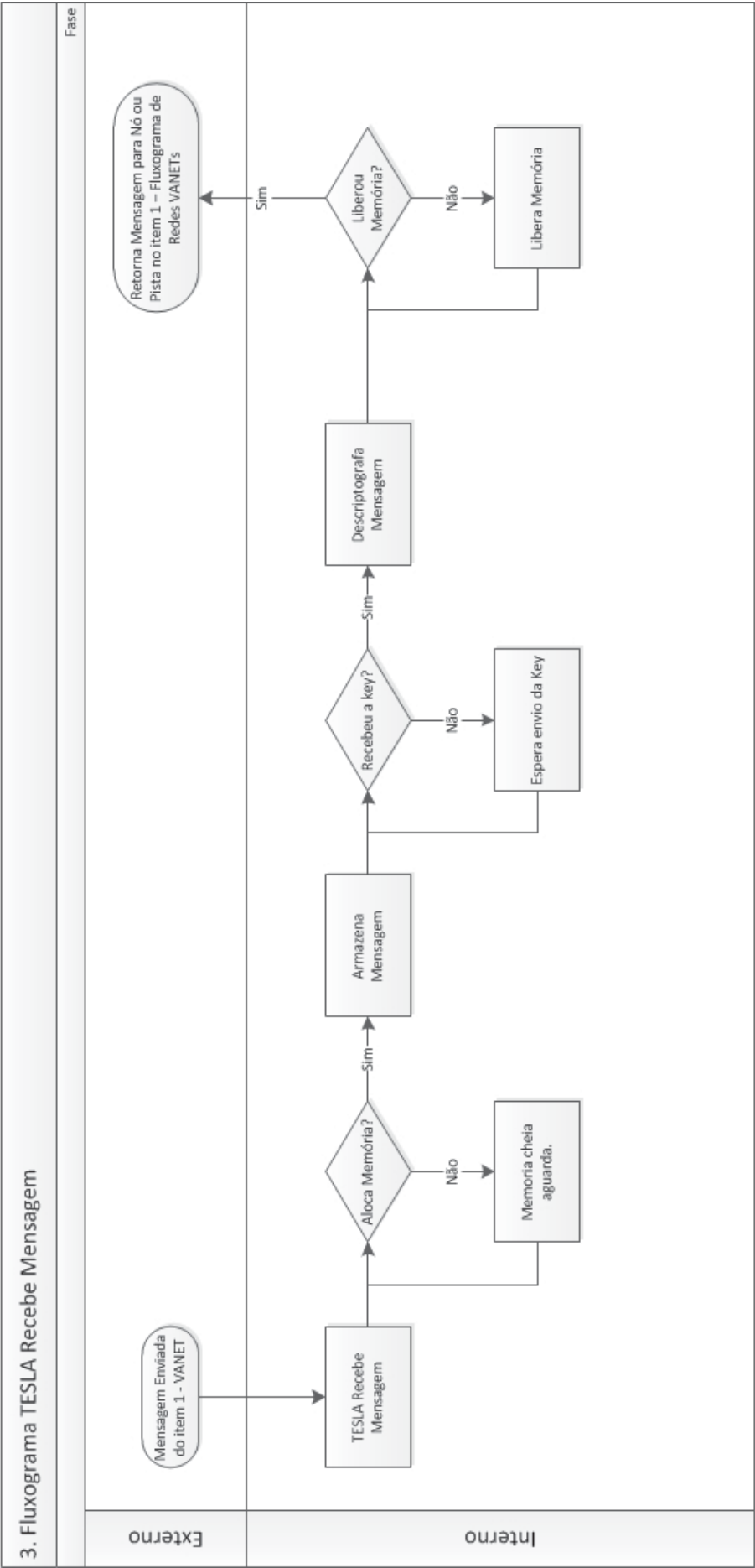


Figura 26 - Fluxograma TESLA Recebe Mensagem

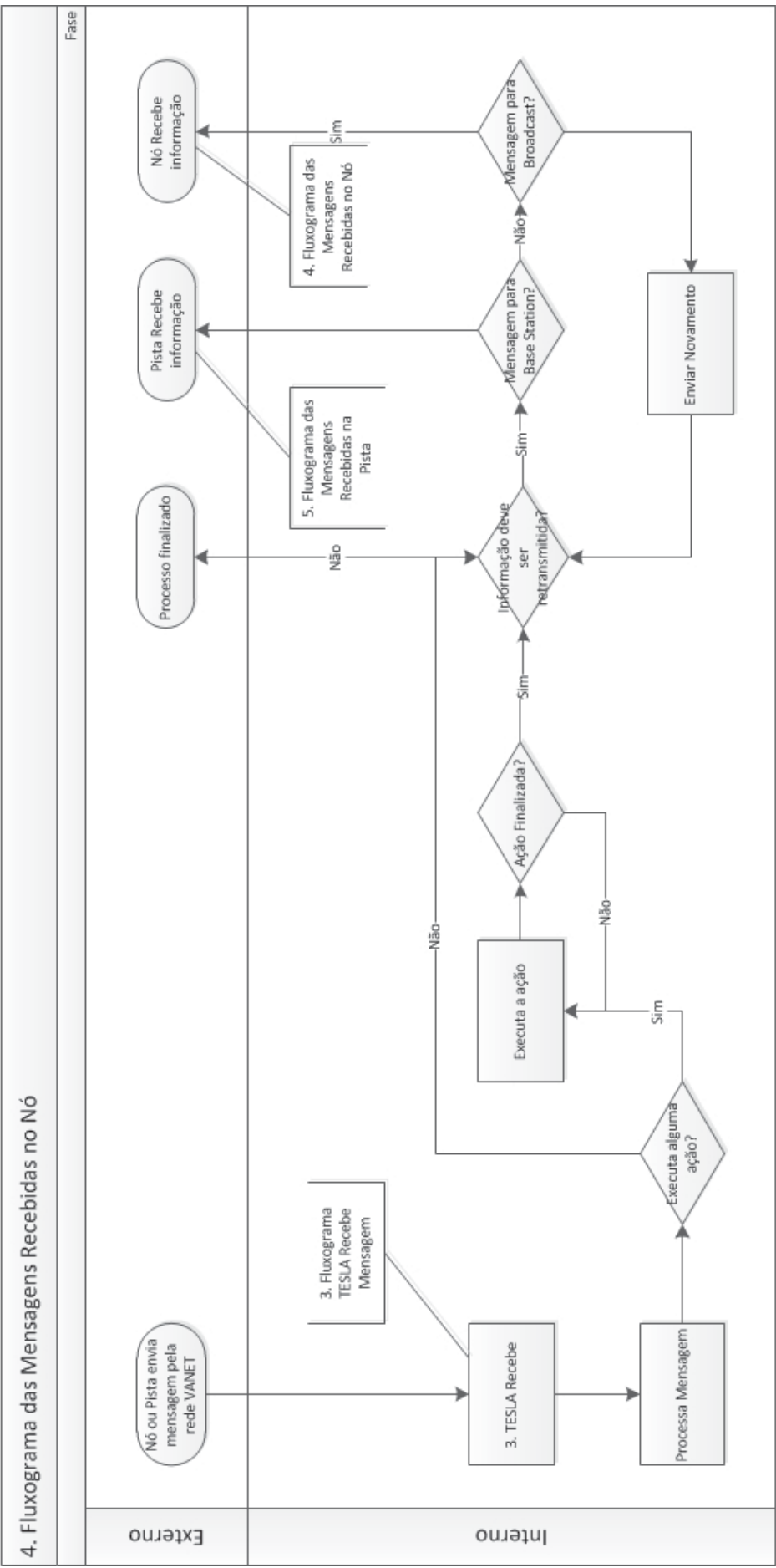


Figura 27 - Fluxograma das Mensagens Recebidas no Nó

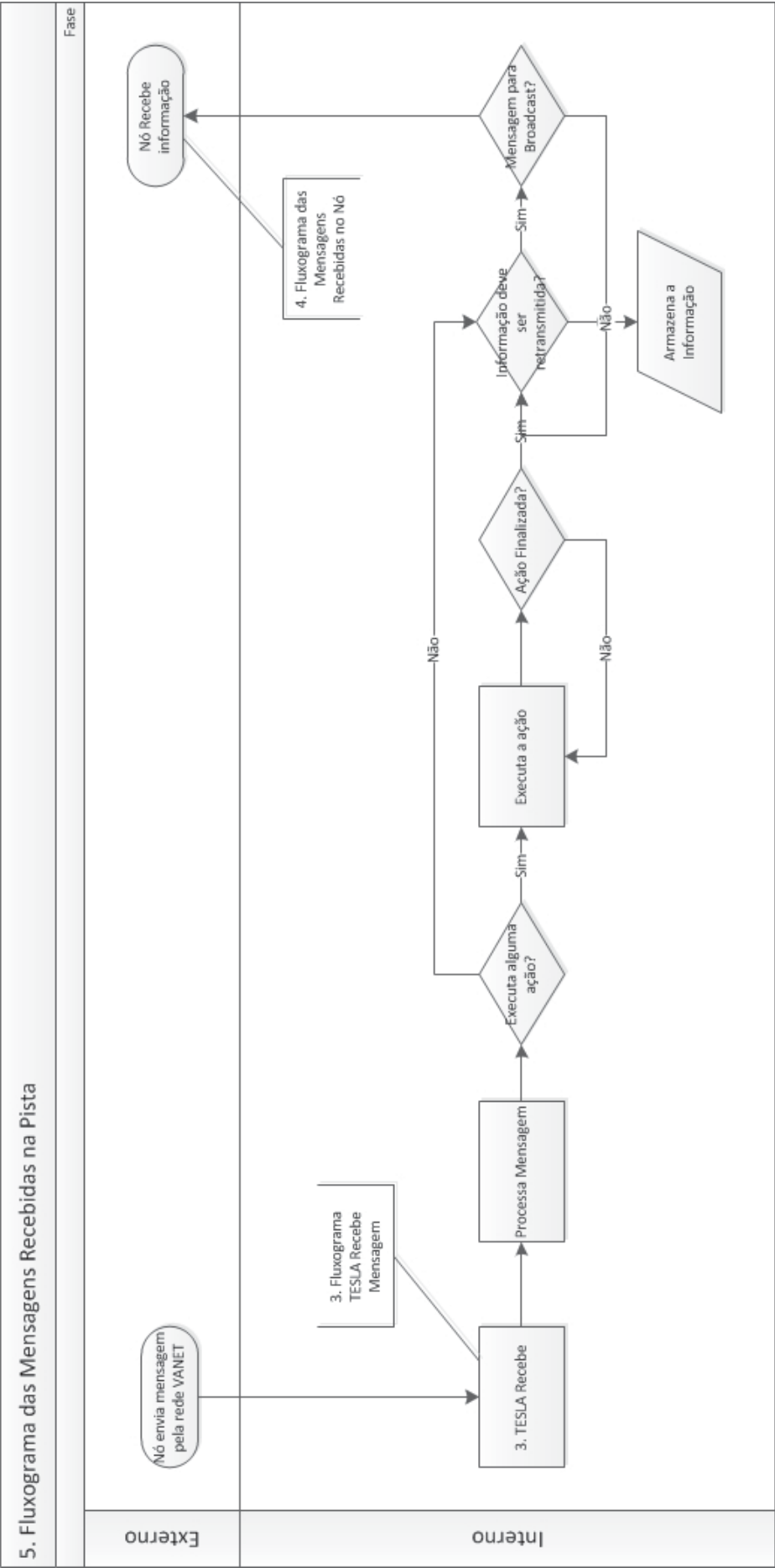


Figura 28 - Fluxograma das Mensagens Recebidas na Pista

4.5 Estrutura e desenvolvimento

O desenvolvimento segundo a modelagem apresentada neste trabalho está estruturado em três partes, o desenvolvimento da criptografia ECC, a alteração do protocolo TESLA para considerar a opção de escolha da criptografia que vai ser utilizada e a alteração na rede VANET para ter a opção de escolha do protocolo que será utilizada no mesmo.

No decorrer dos estudos deste trabalho foram encontrados diversos fatores e detalhes que impossibilitaram o desenvolvimento de duas das três partes, em virtude da complexidade do desenvolvimento de seus detalhes. Não foi possível desenvolver a alteração no protocolo TESLA e a alteração na rede VANET, mas foi possível o desenvolvimento da criptografia ECC.

No decorrer do estudo o desenvolvimento foi realizado na linguagem de programação C++ com a ferramenta Visual Studio 2010 Ultimate, com o Microsoft.Net Framework 4 em ambiente Windons XP Professional com Service Pack 3, com hardware Intel Core 2 Duo T7250 2GHz e com 3 GB de Ram.

O desenvolvimento seguiu o diagrama de classe e o cálculo apresentado na seção 3.2.2. Sua implementação seguiu a modelagem apresentada neste estudo, como não é o objetivo deste trabalho não será detalhado o fonte deste desenvolvimento apenas o resultado final do mesmo.

A Figura 29 mostra o retorno da chamada da classe EllipticCurve para a realização da criptografia de uma palavra com 5 letras e a Figura 30 mostra o retorno da descriptografia realizada utilizando a mesma classe EllipticCurve e com a mesma palavra.

```
Ponto Criptografado = <Pa,c1,c2> = <<3, 89>, 60, 37>
Ponto Criptografado = <Pa,c1,c2> = <<195, 256>, 51, 45>
Ponto Criptografado = <Pa,c1,c2> = <<221, 33>, 103, 244>
Ponto Criptografado = <Pa,c1,c2> = <<15, 231>, 110, 153>
Ponto Criptografado = <Pa,c1,c2> = <<228, 152>, 202, 3>
```

Figura 29 - Retorno da Classe EllipticCurve (Criptografia)

```
Ponto Descriptografado = <134, 234>  
Ponto Descriptografado = <75, 238>  
Ponto Descriptografado = <40, 55>  
Ponto Descriptografado = <40, 55>  
Ponto Descriptografado = <194, 35>
```

Figura 30 - Retorno da Classe EllipticCurve (Descriptografia)

4.6 Considerações Finais

Neste Capítulo, foi apresentada a modelagem sugerida na proposta deste estudo na forma de modelagem UML e modelagem Fluxograma, baseadas no conceito apresentado no capítulo 1, para futuros estudos e desenvolvimento.

Foi baseada a modelagem no modelo UML onde foi dividido em diagrama classe, diagrama de case, digrama de sequência e diagrama de estado onde pode-se ter a visão geral das alterações.

Ainda neste capítulo foi apresentado o desenvolvimento realizado durante os estudos, sendo ele o desenvolvimento da criptografia ECC e sua descriptografia.

No próximo capítulo serão apresentadas as considerações finais deste estudo e possíveis trabalhos futuros.

5. Conclusão e Trabalhos Futuros

Nesse capítulo são apresentadas as considerações finais deste estudo e em seguida possíveis trabalhos futuros.

5.1 Conclusão

O aumento constante dos acidentes, do tráfego, dos congestionamentos, dos roubos de dados nas rodovias, nas cidades e dos ataques nas redes por pessoas mal intencionadas, fizeram com que se intensifica os estudos nas redes VANETs que se encontra em evidência por se tratar de uma rede específica para evitar estes problemas.

Mas a rede VANET tem como função principal ajudar e direcionar o trânsito e evitar os congestionamentos, por isso, não passa despercebido pelas pessoas mal intencionadas que, por vários motivos como ganhar prioridade nas vias ou pelo simples fato de mostrar que consegue invadir a rede e/ou para roubar dados de outros integrantes da mesma entre outros motivos, acabam sendo alvos de invasões.

Para solucionar esta dificuldade foi modelada uma proposta de alteração na comunicação da segurança das redes VANETs e com isto surgiu a necessidade da utilização de um novo protocolo de segurança sendo este o protocolo TESLA. Portanto, a modelagem inclui o protocolo TESLA como opção de segurança para dar suporte na troca dos dados entre os nós/pista na rede VANET.

Porém, ainda há falhas nas redes devido aos ruídos e obstáculos, e estas falhas podem ocasionar a perda de pacotes e consequentemente a falha na entrega da

mensagem. Outro fato relevante é que o espaço de memória em um veículo não é limitado como acontece na rede Ad Hoc, contudo não podemos descartar este item, pois no protocolo TESLA é necessário guardar na memória a mensagem inteira enviada até a chegada da chave. Muitas pessoas mal intencionadas utilizam desta necessidade para fazer ataques, por exemplo, ao enviar diversas mensagens para o mesmo receptor e sobrecarregar a memória do mesmo com mensagens falsas até prejudicar a rede por lentidão ou falta de memória.

Por isso, foi modelada a proposta de alteração da criptografia utilizada no protocolo TESLA, sendo ela ECC que até o momento deste estudo é considerada a mais segura pelo seu cálculo difícil e a utilização de chaves curtas.

Para um bom funcionamento da criptografia com o protocolo TESLA é necessário configurar os parâmetros da ECC levando em conta as características da rede VANET, com isso o ganho de processamento pode ser considerável.

Diante da modelagem apresentada neste trabalho foi implementado o algoritmo da criptografia ECC em linguagem de programação C++ com a ferramenta Visual Studio 2010 Ultimate com o Microsoft .Net Framework 4 em ambiente Windows XP Professional com Service Pack 3 com hardware Intel Core 2 Duo T7250 2GHz com 3 GB de Ram. Pela complexidade e dificuldade apresentada na rede VANET e no protocolo TESLA não foi possível seu desenvolvimento neste trabalho.

Portanto, pela complexidade no desenvolvimento não foi possível a implementação neste trabalho, sendo assim, não é possível afirmar que a rede VANET com estas modificações seria mais segura e confiável que a rede VANET em seu estado original. Porém, conceitualmente, com os itens apresentados neste trabalho pode-se concluir que esta modificação terá resultados significativos na segurança.

5.2 Trabalhos Futuros

Para trabalhos futuros tem-se o desenvolvimento da alteração no protocolo TESLA para ser possível a escolha de qual criptografia deseja ser utilizada.

Na rede VANET tem-se a alteração para ser possível a escolha de qual protocolo de segurança deseja-se utilizar.

Ainda no protocolo TESLA verificar a opção de utilizar o TESLA++ na rede VANET, para um melhor aproveitamento da memória no receptor, com isso obter

um ganho de desempenho e evitar possíveis ataques conforme apresentado anteriormente neste estudo.

6. Referências Bibliografia

BAJAJ, L. et al. **GloMoSim: a scalable network simulation environment**. Los Angeles, 1999. Disponível em: <
<http://www.google.com.br/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0CEEQFjAB&url=http%3A%2F%2Fciteseerx.ist.psu.edu%2Fviewdoc%2Fdownload%3Fdoi%3D10.1.1.101.2658%26rep%3Drep1%26type%3Dpdf&ei=8SyXUJ7RC8bL0AHyyoC4DA&usg=AFQjCNGPVWbfXWulLErzdB3htkFkRIpV6A>> Acessado em: 5 Out. 2012.

BARBOSA, J. C. **Criptografia de chave pública baseada em curvas elípticas**. 2003. 28 f. Dissertação (Mestrado em Redes) – COPPE – Instituto Alberto Luiz Coimbra de Pós-graduação e Pesquisa de Engenharia, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2003. Disponível em:
 <<http://www.lockabit.coppe.ufrj.br/downloads/academicos/ECCMono.pdf>>. Acesso em: 13 Nov. 2010.

BONEH, D.; FRANKLIN, M. **Identity-Based Encryption from the Weil Pairing**. Proceedings of Crypto 2001. Disponível em:
 <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.66.1131&rep=rep1&type=pdf>>. Acesso em: 13 Nov. 2010.

BOUKERCHE, A.; YONGLIN, R.; PAZZI, R. W. N. **An Adaptive Computational Trust Model for Mobile *Ad Hoc* Networks** – Proceedings of the ACM IWCMC – pp. 191-195 – 2009.

CHAURASIA, B.K.; VERMA, S.; TOMAR, G.S. **Attacks on Anonymity in VANET**, Computational Intelligence and Communication Networks (CICN), 2011 International Conference on Digital Object Identifier, 2011, Página(s): 217 – 221 Disponível em
 <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6112858>> Acessado em: 20 Out. 2012.

CHEUNG, R.C.C. et al. **Customizable elliptic curve cryptosystems**. Sept/2005.

Disponível em: <

http://ieeexplore.ieee.org/search/freesrchabstract.jsp?tp=&arnumber=1525037&queryText%3DCustomizable+Elliptic+Curve+Cryptosystems%26openedRefinements%3D*%26searchField%3DSearch+All> Acessado em: 13 Mai. 2011.

CHOI, J.; JUNG, S. **A Security Framework with Strong Non-repudiation and Privacy in VANETs**. 6th IEEE. Consumer Communications and Networking

Conference, 2009. CCNC 2009., Las Vegas. Disponível em:

<http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4784730>. Acessado em: 13 Nov. 2010.

CHOI, H. M.; HONG, C. P.; KIM, C. H. **High Performance Elliptic Curve**

Cryptographic Processor Over GF(2¹⁶³), Electronic Design, Test and Applications,

2008. DELTA 2008. 4th IEEE International Symposium on Jan/2008. Disponível em: <

http://ieeexplore.ieee.org/search/freesrchabstract.jsp?tp=&arnumber=4459559&queryText%3DHigh+Performance+Elliptic+Curve+Cryptographic+Processor+Over+GF.LB.2163+.RB.%26openedRefinements%3D*%26searchField%3DSearch+All> Acessado em: 13 Mai. 2011.

FALL, K. E VARADHAN, K. **The ns Manual (formerly ns Notes and Documentation)** – The VINT Project – 2002.

FOURNARIS, A. P.; KOUFOPAVLOU, O. **Creating an Elliptic Curve Arithmetic Unit for use in Elliptic Curve Cryptography**, Sept/2008. Disponível em: <

http://ieeexplore.ieee.org/search/freesrchabstract.jsp?tp=&arnumber=4638588&queryText%3DCreating+an+Elliptic+Curve+Arithmetic+Unit+for+use+in+Elliptic+Curve+Cryptography%26openedRefinements%3D*%26searchField%3DSearch+All> Acessado em: 13 Mai. 2011.

IEEE SECURITY AND PRIVACY – DOI 10.1109/MSP.2004.26 – 2004.

JANECH, J.; LIESKOVSKY, A.; KRSAK, E. **Comparison of Strategies for Data Replication in VANET Environment**, Advanced Information Networking and

Applications Workshops (WAINA), 2012 26th International Conference on. Digital Object Identifier, 2012 , Page(s): 575 – 580. Disponível em

<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6185324>> Acessado em: 20 Out. 2012.

KARNADI, F. K.; MO, Z. H.; LAN, K. C. **Rapid generation of realistic mobility models for VANET**, IEEE Wireless Communications and Networking

Conference (WCNC) – 2506–2511 – 2007.

KRAJZEWICZ, D. et al. **SUMO (Simulation of Urban MObility) an open-source traffic simulation** – Middle East Symposium on Simulation and Modelling (MESM) – 183–187 – 2002.

LIN, X. et al. **Performance enhancement for secure vehicular communications**.

IEEE *Global Telecommunications Conference, GLOBECOM '07*, Washington, 2007.

Disponível em: <http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4411006>.
Acesso em: 13 Nov. 2010.

MAJKOWSKI, P. et al. **Heterogenic Distributed System for Cryptanalysis of Elliptic Curve Based Cryptosystems**, Aug/2008. Disponível em: <http://ieeexplore.ieee.org/search/freesrchabstract.jsp?tp=&arnumber=4616654&queryText%3DHeterogenic+Distributed+System+for+Cryptanalysis+of+Elliptic+Curve+Based++Cryptosystems%26openedRefinements%3D*%26searchField%3DSearch+All>
Acessado em: 13 Mai. 2011.

MARQUES, L. G. **Curvas elípticas: aplicações criptográficas**. 2007. 94 f. Trabalho de conclusão de curso (Bacharelado)-Universidade Federal de Goiás, Catalão, 2007. Disponível em: <<http://www.catalao.ufg.br/cc/disc/pfc/mono/Leonardo2007.pdf>>. Acesso em: 3 Out. 2012.

MATOS, J. A. **Implementação e avaliação de desempenho de software livre para computação científica em ambiente paralelo**. 2008, 100 f. Dissertação (Mestrado em Ciência em Engenharia Elétrica) – UFRJ – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2008. Disponível em: <<http://www.pee.ufrj.br/teses/textocompleto/2008032801.pdf>> Acessado em: 13 Mai 2011.

MENDES, A. V. **Estudo de criptografia com chave pública baseada em curvas elípticas**. 2007. 49 f. Monografia (Bacharel em Sistemas de Informação) – Unimontes – Universidade Estadual de Montes Claros, Montes Claros, 2007. Disponível em: <<http://www.ccet.unimontes.br/arquivos/monografias/261.pdf>> Acessado em: 13 Mai 2011.

MENEGUETTI, R. **Rede overlay de suporte a ambientes virtuais colaborativos em redes veiculares**. 2009. 108 f. Dissertação (Mestrado em Ciência da Computação) – Ufscar – Pós-graduação em Ciência da Computação, Universidade Federal de São Carlos, São Carlos, 2009. Disponível em: <http://www.bdt.d.ufscar.br/htdocs/tedeSimplificado//tde_busca/arquivo.php?codArquivo=2938>. Acesso em: 13 Nov. 2010.

MOLVA, R.; SAMFAT, D.; TSUDIK, G. **Authentication in mobile networks**. IEEE Network, March/April 1994. Internet Public Key Infrastructure, Internet RFCs 2459, 2510, 2511. Disponível em: <<http://www.eurecom.fr/~nsteam/Papers/mob.pdf>>.
Acesso em: 13 Nov. 2010.

OLIVEIRA, R. R.; WANGHAM, M. S. **Comparativo entre algoritmos de roteamento utilizando em redes veiculares**. Publicado: Congresso Sul Brasileiro de Computação, 2010, Criciúma. Anais do V Congresso Sul Brasileiro de Computação, 2010. p. 300-307. Disponível em: <<http://periodicos.unesc.net/index.php/sulcomp/article/viewFile/300/307>> Acessado em: 13 Jun 2011.

Personal and Ubiquitous Computing V.11:505–506 – DOI 10.1007/s00779-006-0113-4 – 2007.

Personal and Ubiquitous Computing V.13:445–447 – DOI 10.1007/s00779-009-0230-y – 2009.

Personal and Ubiquitous Computing V.13:389–390 – DOI 10.1007/s00779-008-0210-7 – 2009.

PIORKOWSKI, M., et al.; **TraNS: realistic joint traffic and network simulator for VANETs** – ACM SIGMOBILE Mobile Computing and Communications Review – 12(1):31–33 – 2008.

RADU, D. et al. **Acoustic noise pollution monitoring in an urban environment using a VANET network**. Automation Quality and Testing Robotics (AQTR), 2012 IEEE International Conference on Digital Object Identifier, 2012 , Página(s): 244 – 248. Disponível em <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6237711>> Acessado em 20 Out. 2012.

RAYA, M.; HUBAUX, J. P. **The security of vehicular Ad Hoc networks**. SASN 2005, Alexandria, VA, USA – 2005.

RODRIGUES, D. M. C. **Implementação da cifra de curva elíptica num PDA**. 2008. 93 f. Dissertação (Mestrado Integrado em Engenharia Electrotécnica e de Computadores) – Universidade Técnica de Lisboa, 2009. Disponível em: <<https://dspace.ist.utl.pt/bitstream/2295/281570/1/Tese.Diogo.Rodrigues.MEEC.L50702.pdf>> Acessado em: 13 Mai 2011.

RUAN, N.; HORI, Y. **DoS attack-tolerant TESLA-based broadcast authentication protocol in Internet of Things**. Mobile and Wireless Networking (iCOST), 2012 International Conference on Selected Topics in Digital Object Identifier, 2012, Página(s): 60 – 65.

SAVARI, M.; MONTAZEROLZOHOUR, M.; YEOH E. T. **Comparison of ECC and RSA algorithm in multipurpose smart card application**, Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), 2012 International Conference on Digital Object Identifier, 2012 , Página(s): 49 – 53.

SCHNEIER, B. **Applied cryptography: protocols, algorithms, and source code in C**. 2. ed. New York: John Wiley & Sons, 1996.

SHOUZHI, X. et al. **An improved sliding window algorithm for ECC multiplication**. World Automation Congress (WAC), 2012, Página(s): 335 – 338. Disponível em <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6321291>> Acessado em 20 Out. 2012.

SHOUZHI X. et al. **Simulated study on video communication over VANET**. World Automation Congress (WAC), 2012, Página(s): 221 – 225. Disponível em <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6321063>> Acessado em 20 Out. 2012.

SICHITIU, M. L.; KIHLM. **Inter-Vehicle Communication Systems: A Survey**, 2ND Quarter 2008, Volume 10, N° 2, 2008.

STALLINGS, W. **Criptografia e Segurança de Redes: Princípios e Práticas**. 4. ed. São Paulo: Pearson, 2008.

STUDER, A. et al. **Full paper: flexible, extensible, and efficient VANET authentication**. Março/2008. Disponível em: < http://www.cylab.cmu.edu/files/pdfs/tech_reports/cmucylab08010.pdf>. Acesso em: 13 Nov. 2011

SUNNADKAL, R.; SOH, B.; PHAN, H. **A four-stage design approach towards securing a vehicular Ad Hoc networks architecture**, IEEE Network, Janeiro/2010. Disponível em: <<http://ieeexplore.ieee.org/Xplore/login.jsp?url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F5438527%2F5438652%2F05438694.pdf%3Farnumber%3D5438694&authDecision=-203>>. Acesso em: 13 Nov. 2010.

SUTIKNO, S.; SURYA, A.; EFFENDI, R. **An implementation of elGamal elliptic curves cryptosystems**. Circuits and Systems, 1998. IEEE APCCAS 1998. The 1998 IEEE Asia-Pacific Conference em Nov/1998. Disponível em: < http://ieeexplore.ieee.org/search/freesrchabstract.jsp?tp=&arnumber=743829&queryText%3DAn+Implementation+of+ElGamal+Elliptic+Curves+Cryptosystems%26openedRefinements%3D*%26searchField%3DSearch+All> Acessado em: 13 Mai. 2011.

TAYAL, S.; TRIPATHY, M.R. **VANET - challenges in selection of vehicular mobility model**. Advanced Computing & Communication Technologies (ACCT), 2012 Second International Conference on Digital Object Identifier, 2012 , Página(s): 231 – 235. Disponível em <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6168367>> Acessado em 20 Out. 2012.

VIGILA, S.; MUNEESWARAN, K. **Implementation of text based cryptosystem using elliptic curve cryptography**. Advanced Computing, 2009. ICAC 2009. First International Conference on Dec/2009. Disponível em: < http://ieeexplore.ieee.org/search/freesrchabstract.jsp?tp=&arnumber=5378025&queryText%3DImplementation+of+Text+based+Cryptosystem+using+Elliptic+Curve+Cryptography%26openedRefinements%3D*%26searchField%3DSearch+All> Acessado em: 13 Mai. 2011.

WANG, M.; DAI, G. **An efficient generation method of elliptic curve for pairing - based cryptosystems**. IEEE Network, Abril/2010. Disponível em: < http://ieeexplore.ieee.org/search/freesrchabstract.jsp?tp=&arnumber=5532701&queryText%3DAn+Efficient+Generation+Method+of+Elliptic+Curve+for+Pairing-based+Cryptosystems%26openedRefinements%3D*%26searchField%3DSearch+All> Acessado em: 13 Mai. 2011.

WASEF, A.; SHEN, X. **MAAC: message authentication acceleration protocol for vehicular Ad Hoc networks**. IEEE Network, Dezembro/2009. Disponível em: <http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5425220>. Acesso em: 13 Nov. 2010.

WASHINGTON, L.C. **Elliptic Curves: number theory and cryptography**. 2. ed. Boca Raton. Chapman & Hall/CRC. 2008.

YAN T.; ZHANG W.; WANG G. **DOVE: data dissemination to a fixed number of receivers in VANET**. Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2012 9th Annual IEEE Communications Society Conference on Digital Object Identifier, 2012 , Página(s): 272 – 280. Disponível em <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6275787>> Acessado em 20 Out. 2012.

YILIANG, H. et al. **Verifiable threshold cryptosystems based on elliptic curve**. Computer Networks and Mobile Computing, 2003. ICCNMC 2003. 2003 International Conference on Oct/2003. Disponível em: <http://ieeexplore.ieee.org/search/freesrchabstract.jsp?tp=&arnumber=1243064&queryText%3DVerifiable+Threshold+Cryptosystems+Based+on+Elliptic+Curve%26openedRefinements%3D*%26searchField%3DSearch+All> Acessado em: 13 Mai. 2011.

YONGLIN, R; BOUKERCHE, A.; **An Efficient Trust-based Reputation Protocol for Wireless and Mobile Ad hoc Networks: Proof and Correctness** – Proceedings of the IEEE GlobeCom – 2008.

YU, B. **Establishment of elliptic curve cryptosystem**. Information Theory and Information Security (ICITIS), 2010 IEEE International Conference on Dec/2010. Disponível em: <http://ieeexplore.ieee.org/search/freesrchabstract.jsp?tp=&arnumber=5689767&queryText%3DEstablishment+of+Elliptic+Curve+Cryptosystem%26openedRefinements%3D*%26searchField%3DSearch+All> Acessado em: 13 Mai. 2011.

ZARKI, M. E. et al. **Security issues in a future vehicular network**. in Proc. European Wireless 2002, Feb. 2002. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.122.2138&rep=rep1&type=pdf>>. Acesso em: 13 Nov. 2010.

ZIED, G.; MOHSEN, M.; RACHED, T. **On the hardware design of elliptic curve public key cryptosystems using programmable cellular automata**. Signals, Circuits and Systems, 2008. SCS 2008. 2nd International Conference on Nov/2008. Disponível em: <http://ieeexplore.ieee.org/search/freesrchabstract.jsp?tp=&arnumber=4746947&queryText%3DOn+the+hardware+design+of+elliptic+curve+public+key++cryptosystems+using+programmable+cellular+automata%26openedRefinements%3D*%26searchField%3DSearch+All> Acessado em: 13 Mai. 2011.

ZUXIU X. **A modified security teach-lab management model based on ECC digital certificate**. Intelligent Computation Technology and Automation (ICICTA), 2012 Fifth International Conference on Digital Object Identifier, 2012 , Página(s): 18 – 21. Disponível em <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6150226>> Acessado em 20 Out. 2012.

APÊNDICE A – Código do desenvolvimento das ECC

O desenvolvimento neste trabalho foi das Curvas Elípticas, que serão apresentados a seguir:

Valida Número

```

1- function g=validaNumero(a, mo)

2- num = []
3- den = []
4- x = []
5- resto = 1
6- c = 0
7- [num, den] = rat(a)

8- while resto <> 0
9- x = mo*c + num
10- if den <> 0 then
11- resto = x/den
12- else
13- resto = x
14- end
15- resto = resto - int(resto)
16- c = c + 1
17- end

18- if den <> 0 then
19- g = x/den
20- else
21- g = x
22- end

23- endfunction

```

Modulo

```

1- function Retorno=Modulo(delta, mo)

2- while delta >= mo
3-   delta = delta - mo
4- end

5- while (delta <= -mo)
6-   delta = delta + mo
7- end

8- if delta < 0 then
9-   delta = delta + mo
10- end

11- Retorno = delta

12- endfunction

```

Soma

```

1- function matriz=soma(a, b, c, d, mo)

2- if (d <> 0) then
3-   //  $(y^2 - y1)/(x2 - x1)$ 
4-   if (c - a) == 0 then
5-     delta = (d - b)
6-   else
7-     delta = (d - b)/(c - a)
8-   end

9-   exec validaNumero.sci
10-   delta = validaNumero(delta, mo)

11- exec modulo.sci
12- delta = Modulo(delta, mo)

13- //  $x^3 = delta^2 - x1 - x2$ 
14- x = (delta^2) - a - c

15- x = Modulo(x, mo)

16- //  $y^3 = delta * (x1 - x3) - Y1$ 
17- y = delta * (a - (x)) - b
18- y = Modulo(y, mo)

19- matriz = [x, y]
20- else

```

```

21- matriz = [a,b]
22- end
23- endfunction

```

Calcula

```

1- function g=calcula(a, x, y)

2- //lista que vai ficar os valores de x e y
3- lista = []
4- //variável que controla se soma ou se multiplica.
5- control = []
6- entra = 0
7- // b recebe a para saber qual momento que deve parar o calculo no while.
8- b = a
9- // mo = valor do modulo que vai ser utilizado no cálculo.
10- mo = 257
11- // este while verifica se o numero passado deve ser somado ou dividido,
12- // estes números são colocados em uma lista para ser utilizado no parte de baixo
   deste fonte
13- // na lista fica 0 e 1, 0 para quando for soma e 1 para quando for multiplicação.
14- while a > 1
15- div = a/2
16- div = div - int(div)
17- if div > 0 then
18- control = [control, 0]
19- a = a - 1
20- else
21- control = [control, 1]
22- a = a/2
23- end
24- end

25- //a variável que vai ser utilizada para controle no while.
26- a = 1

27- // carrega os valores de x e y nestas variáveis para não perder o valor de x e y.
28- // no fonte utilizamos novX e novY.
29- novX = x
30- novY = y

31- while a > b
32- // multiplica os pontos
33- if control(length(control) - entra) == 1 then
34- // carrega a função multiplica
35- exec multiplica.sci
36- //faz a chamada da função multiplicação passando o ponto x e y e o valor de
   modulo.
37- lista = multiplica(novX,novY, mo)

```

```

38- //variável para controle do while
39- a = a * 2
40- else
41- //carrega a função soma
42- exec soma.sci
43- //faz a chamada da função soma passando os pontos x inicial, y inicial, x final,
   y final e o valor do modulo.
44- lista = soma(x,y,novX,novY,mo)
45- // variável de controle do while
46- a = a + 1
47- end
48- // coloca o valor final de x e y novamente nas variáveis para passar novamente
   pelo while
49- novX = lista(1)
50- novY = lista(2)
51- // entra utilizada controlar se soma ou multiplica.
52- entra = entra + 1
53- end
54- // retorna o ponto x e y final.
55- g = lista
56- endfunction

```

Multiplica

```

1- function matriz=multiplica(a, b, mo)

2- if (b <> 0) then
3- // delta = 3*x1^2 + a / 2*y1
4- delta = ((3*(a^2)) + 1)

5- exec modulo.sci

6- delta = Modulo(delta,mo)

7- if (2*b) == 0 then
8- delta = delta
9- else
10- delta = delta/(2*b)
11- end

12- exec validaNumero.sci

13- delta = validaNumero(delta, mo)

14- delta = Modulo(delta, mo)

15- // x^3 = delta^2 - x1 - x2
16- x = (delta^2) - (2*a) // - a - a

```

```

17- x = Modulo(x,mo)

18- //  $y^3 = \text{delta} * (x1 - x3) - y1$ 
19- y = delta*(a-x)-b

20- y = Modulo(y,mo)

21- matriz = [x,y]
22- else
23- matriz = [0,0]
24- end
25- endfunction

```

Criptografa

```

1- function matriz=criptografa(a)

2- //key (qualquer número menor que p)
3- k = 13
4- // key (privada menor p)
5- nb = 17

6- //ponto da curva
7- g = [0,1]

8- // número da tabela asc
9- asc = ascii(a)
10- // segundo ponto da curva
11- h = [1,15]

12- //cria a função calcula
13- exec calcula.sci
14- // começa os cálculos
15- pb = calcula(nb,g(1),g(2))

16- pml = calcula(asc,h(1),h(2))

17- kpb = calcula(k,pb(1),pb(2))
18- //cria a função soma
19- exec soma.sci

20- pmlkpb = soma(pml(1),pml(2),kpb(1),kpb(2),37)

21- kg = calcula(k,g(1),g(2))
22- // retorna os pontos.
23- cm = [kg,pmlkpb]

24- matriz = cm
25- endfunction

```

APÊNDICE B – Diagrama de Estado Completo da Melhoria

Pelo seu tamanho e não tendo como ficar no corpo do texto será apresentada na modelagem UML, o diagrama de estado de forma completa e detalhada da melhoria apresentada no capítulo 4 no apêndice A sendo utilizada a ferramenta DIA para seu desenvolvimento.

O mesmo encontra-se subdividido no capítulo 4 e explicado no detalhe cada particularidade.

