Avaliação de algoritmos de assinatura digital em redes veiculares utilizando ambiente emulado

Diego V. Natividade, Luiz H. A. Correia

Departamento de Ciência da Computação – Universidade Federal de Lavras (UFLA)
Caixa Postal 3037 – 37.200-900 – Lavras – MG – Brasil

natividade@bol.com.br, lcorreia@ufla.br

Abstract. *Advanced Driver Assistance Systems (ADAS) are being embedded in vehicles to reduce the number of accidents. Annually, 1.35 million people die in traffic accidents, so the VANET (Vehicle Ad Hoc Network) will be the next ADAS generation, providing communication and traffic safety. However, malicious users can forge messages on the network. The digital signature algorithms RSA, ECDSA and Ed25519 can be used to guarantee the authenticity of messages. This article evaluates these algorithms in VANET, using an emulated environment. The results show that the Ed25519 is faster than the others and its use is computationally feasible in VANETs, considering the driver's reaction time and the braking distance.*

Resumo. *Os sistemas de assistência ao motorista (ADAS) estão sendo embarcados nos veículos para reduzir o número de acidentes. Anualmente, 1,35 milhões de pessoas morrem em acidentes de trânsito. Com isso, a VANET (Vehicle Ad Hoc Network) vem para prover mais comunicação e segurança no trânsito. Porém, usuários mal intencionados podem forjar mensagens na rede. Os algoritmos de assinatura digital RSA, ECDSA e Ed25519 podem ser usados na garantia de autenticidade das mensagens. Este artigo avalia estes algoritmos em VANET, utilizando um ambiente emulado. Resultados mostram que o Ed25519 é mais rápido que os demais e seu uso é computacionalmente viável em VANETs, considerando o tempo de reação do motorista e a distância de frenagem.*

1. INTRODUÇÃO

O número de veículos no mundo tem se aproximado de quase 1,2 bilhões, em torno de quase um veículo para cada sete habitantes, e no Brasil a frota de veículos vem crescendo a uma taxa de 2,3% ao ano. Com o aumento do número de veículos, o número de acidentes também cresceu significativamente. Estima-se que todos os anos, 1,35 milhões de pessoas morrem no mundo em acidentes de trânsito [World Heath Organization 2019]. No Brasil, dados do Conselho Federal de Medicina (CFM) mostram que a cada hora, cinco pessoas morrem de acidentes de trânsito. Ainda revelam que entre 2008 e 2016, 368.821 pessoas morreram no trânsito nas estradas e ruas do país e deixaram mais de 1,6 milhão de feridos.

Com o intuito de reduzir os acidentes veiculares, diversos fabricantes de veículos têm investido em tecnologias como o ADAS (*Advanced Driver Assistance Systems*), que são dispositivos projetados para evitar colisões e para alertar o motorista sobre prováveis perigos. Outros dispositivos de segurança também têm sido desenvolvidos, como controle

de estabilidade e tração, freios ABS, *air-bags*, etc., todos com o intuito de reduzir o risco de acidentes.

As tecnologias ADAS têm evoluído para dispositivos que incorporem a comunicação de diversas fontes, entre elas radares, imagens, comunicação via satélite e redes de comunicação entre veículos. As VANETs (*Vehicle Ad Hoc Networks*) serão a próxima geração de ADAS, que promoverão a comunicação entre veículos, por meio da rede de telefonia celular ou uma infraestrutura de comunicação *ad hoc* sem fio, fornecendo a troca de informações entre os motoristas e proporcionando mais conforto e segurança no trânsito.

Muitas soluções para a comunicação em VANETs têm sido propostas nos últimos anos, mas apenas a implantação de um sistema para troca de mensagens no trânsito não torna o trânsito mais seguro. Assim, surge um outro problema de segurança veicular: o envio de mensagens falsificadas ou forjadas na rede. Um usuário mal intencionado pode injetar mensagens falsificas na rede, comprometendo seu funcionamento e levando o motorista a tomar decisões incorretas ou ser alvo de ataques. Portanto, para garantir a integridade, autenticidade e não repúdio de uma mensagem, utiliza-se assinatura digital, na qual assinaturas geradas com uma chave privada só podem ser validadas pela chave pública correspondente de quem assinou originalmente a mensagem [Barbara 2018].

Para a assinatura digital de mensagens, algoritmos como `RSA` (Rivest-Shamir-Adleman), `ECDSA` (*Elliptic Curve Digital Signature Algorithm*) e `Ed25519` vêm sendo empregados nos últimos anos. Em redes veiculares, o algoritmo `ECDSA` tem sido usado a bastante tempo devido ao seu baixo custo computacional em relação ao `RSA`, pois utiliza chaves menores com o mesmo nível de segurança [Johnson et al. 2001, Manvi et al. 2009, Ravi and Kulkarni 2013]. O `Ed25519` é um algoritmo relativamente novo, de padrão totalmente aberto e leve que tem ganhado espaço para a assinatura digital. Atualmente, os principais sistemas de segurança vêm adotando gradualmente o `Ed25519`, como é o caso do `TLS 1.3` usado em navegadores WEB, do serviço de `SSH`, da `VPN`, dentre outros [IANIX 2019].

Entretanto, o uso de algoritmos de assinatura digital geram *overhead* na transmissão e processamento das mensagens, o que pode aumentar a latência no envio e recepção das mensagens nas redes veiculares. Um alto *overhead* pode também impactar no atraso da transmissão de mensagens de prevenção de acidentes, influenciar no tempo de reação do motorista e na tomada de decisão de sistemas veiculares autônomos.

De forma a analisar o impacto dos algoritmos de assinatura digital de mensagens nas redes veiculares, este trabalho avaliou os algoritmos `RSA`, `ECDSA` e `Ed25519` em um cenário com dois veículos. O veículo mais a frente se acidenta, parando bruscamente e enviando uma mensagem de acidente. O veículo de trás recebe a mensagem e o motorista reage acionando os freios e, consequentemente, parando o automóvel.

Neste cenário, é avaliado todo o tempo gasto, desde a assinatura digital da mensagem (realizada pelo veículo acidentado), o tempo de checagem da assinatura (pelo veículo que recebe a mensagem), bem como os tempos de reação do motorista e frenagem. É obtida então, a distância percorrida pelo veículo de trás, desde o momento do acidente, até sua parada por completo, de acordo com sua velocidade. Este cenário foi emulado em um ambiente híbrido [de Bettio et al. 2019], usando: os simuladores OMNet++ e SUMO para

troca de mensagens e mobilidade de veículos; o *framework* VEINS, que implementa os protocolos WAVE (*Wireless Access in Vehicular Environments*); um hardware embarcado, emulando um dispositivo veicular para computação da assinatura digital da mensagem e sua verificação.

A principal contribuição deste trabalho é avaliar algoritmos de assinatura digital em redes veiculares utilizando ambiente emulado. A ideia central é que haja segurança e autenticidade na troca de mensagens críticas de trânsito,,através de um sistema de assinatura digital, que: seja célere o suficiente para suportar as restrições de tempo de contato entre veículos, seja trivial para rodar em hardwares embarcados e que considere o tempo de reação do motorista e a distância de frenagem.

Este trabalho comparou, no contexto de VANET, três algoritmos de assinatura digital: `RSA`, `ECDSA` e `Ed25519`. Estes foram implementados em `C++` com a biblioteca `Crypto++` em um hardware embarcado, `Gumstix Overo Fire Storm P`[1]. Resultados mostraram que o algoritmo `Ed25519` é mais rápido que os demais e que seu uso é viável em VANET.

Com os testes realizados, as contribuições deste trabalho foram: i) comparar e avaliar os algoritmos de assinatura digital `RSA`, `ECDSA` e `Ed25519` no contexto de redes veiculares; ii) avaliar estes algoritmos em um hardware embarcado; iii) comprovar a viabilidade do algoritmo `Ed25519` no contexto de redes veiculares, levando em consideração o tempo gasto na reação do motorista e a distância de frenagem. Pelo nosso conhecimento, este é o primeiro trabalho a avaliar o desempenho do `Ed25519` em VANETs.

Este artigo está organizado como descrito a seguir. A Seção 2 apresenta os conceitos básicos. Os trabalhos relacionados sobre assinatura digital em VANET são descritos na Seção 3. A metodologia de avaliação é apresentada na Seção 4. A avaliação e discussão dos resultados são mostrados na Seção 5. Por fim, a Seção 6, apresenta as conclusões e trabalhos futuros.

2. Referencial teórico

Vehicle Ad Hoc Network, ou simplesmente VANET é o termo usado para se referir à rede veicular, na qual, cada veículo é equipado com dispositivos denominados OBU (*On-board Unit*), responsáveis pela comunicação sem fio. O termo foi utilizado a princípio para se referir a comunicação direta entre os veículos, comunicação V2V (*Vehicle to Vehicle*), como ilustra a Figura 1(a). No entanto, existe também a troca de mensagens entre veículos e a infraestrutura de rede, comunicação V2I (Fig.1(b)), através de dispositivos localizados na via, chamados de `RSU` (*Road Side Unit*) [Hartenstein and Laberteaux 2008].

O principal objetivo de uma rede veicular é trazer segurança para o motorista, através de compartilhamento de informações da via e das condições de tráfego. Mas pode também ser utilizada para trocar informações de entretenimento, como por exemplo arquivos de multimídia, por meio de conexão com a Internet [Mishra et al. 2016]. Com a introdução de IoT (*Internet of Things*) outro paradigma tem sido utilizado em redes veiculares, o V2X (*Vehicle to Everthing*), que é utilizado para se referir não só a comunicação entre os veículos e a infraestrutura, mas também a comunicação dos veículos com dispositivos, pedestres, e qualquer outro sistema que possam interferir ou afetar os veículos

[1]https://s3-us-west-2.amazonaws.com/media.gumstix.com/datasheets/GUM3703FP.pdf

[Bhover et al. 2017]. A Figura 1 (c) ilustra esta comunicação.

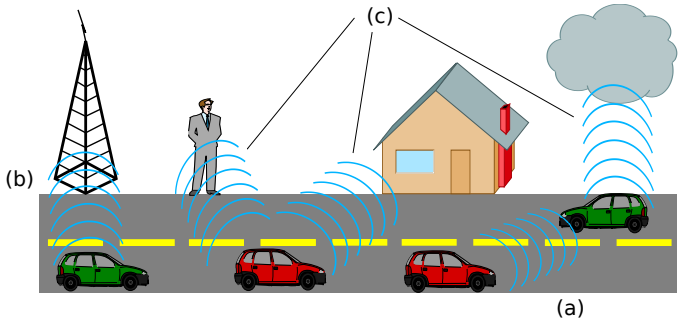

Figura 1. Comunicação V2V (a), V2I (b) e V2X (c).

A assinatura digital é um caso particular de criptografia assimétrica. Segundo [Barker 2016], em sistemas criptográficos, a segurança de um algoritmo é avaliada pelo nível de segurança, ou, o termo mais usado atualmente, `força de segurança`, dado em bits. Isto indica o quanto de trabalho é despendido para se quebrar um algoritmo de criptografia. Para [Kalra and Sood 2011], a força de segurança depende do tipo de algoritmo e a complexidade do problema matemático abordado e do tamanho da chave utilizada, dada também em bits.

A Figura 2 ilustra o procedimento de assinatura digital (a) e verificação de assinatura (b), em que o par de chaves do remetente é usado no processo. Para assinar uma mensagem (1), o remetente gera um *hash* (2) da mensagem e criptografa (3) esse *hash* com sua chave privada, obtendo assim uma mensagem assinada digitalmente (4). Ao receber a mensagem assinada (5), o destinatário usa a chave pública do remetente para verificar a assinatura digital da seguinte maneira: descriptografa o *hash* (6) recebido; gera um novo *hash* (7) da mensagem (usando o mesmo algoritmo de geração de *hash* usado no remetente); finalmente, compara (8) o *hash* gerado com o *hash* descriptografado; se ambos forem iguais, a assinatura será considerada válida.

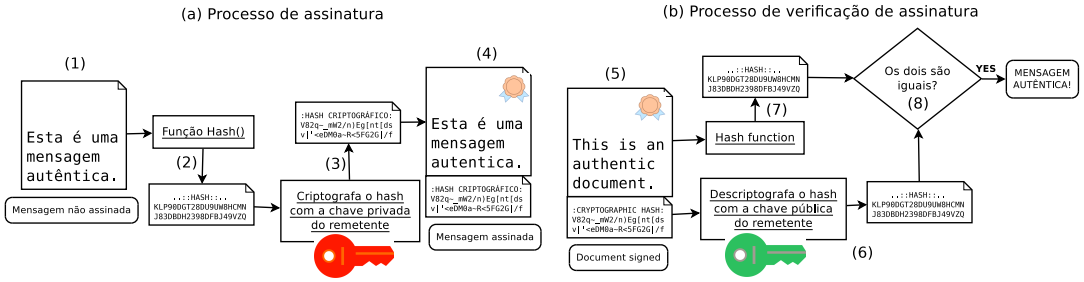

Figura 2. Processo de assinatura digital

Segundo [Barker 2016], um algoritmo de força de segurança igual a 112 bits ainda será considerado seguro até o ano de 2030, e com 128 bits, além do ano de 2031, de acordo com a computação que conhecemos hoje. Existem diversos algoritmos de assinatura digital e que abordam problemas matemáticos distintos. A seguir, serão apresentados os três algoritmos de assinatura digital abordados neste trabalho: `RSA`, `ECDSA` e `Ed25519`.

O algoritmo de criptografia de chave pública chamado RSA foi criado em 1977 por Ron Rivest, Adi Shamir and Leonard Adleman. Trata-se de um algoritmo de IFC (*Integer Factorization Cryptography*) que explora o problema matemático da fatoração do produto de dois números primos grandes [Barker 2016]. Segundo [Perbawa et al. 2017], o tamanho utilizado nas chaves do algoritmo RSA, variam normalmente de 1024 a 2048 bits. A segurança do sistema está na dificuldade de fatorar grandes números [Rivest et al. 1978].

O ECDSA é um algoritmo de ECC (*Elliptical Curve Cryptography* - Criptografia de Curvas Elípticas). Sua segurança é baseada no problema matemático ECDLP (*Elliptic Curve Discrete Logarithm Problem*). Isso aumenta significativamente a força de segurança por bit de chave em um algoritmo que usa curvas elípticas. Por ser um algoritmo ECC, naturalmente o ECDSA possui uma sobrecarga reduzida em comparação ao RSA. A segurança fornecida por uma chave criptográfica ECC de 160 bits é equivalente a uma chave de 1024 bits do algoritmo RSA [Manvi et al. 2009].

Assim como o ECDSA, o Ed25519 é também um algoritmo de curvas elípticas, mas que usa curvas torcidas de Edwards (*twisted* Edwards *curves*) e *hash* SHA-512 [Bernstein 2006, Bernstein et al. 2015, Josefsson and Liusvaara 2017]. É um algoritmo de criptografia relativamente novo e possui um nível de segurança equivalente a 128 bits. Tem esse nome pois é baseado em curvas de Edwards e reduz o campo ao número primo $2^{(255)} - 19$. O algoritmo é eficiente em mensagens curtas devido a forma como foi construído [Bernstein et al. 2012].

A Tabela 1 é uma adaptação de [Bernstein et al. 2012] e [Barker 2016], na qual mostra a comparação de três algoritmos de assinatura digital diferentes, quanto ao seu tamanho de chave e uma força de segurança de aproximadamente 128 bits.

Tabela 1. Tamanho da chave e força de segurança dos algoritmos de assinatura digital

Algoritmo	Tamanho da chave	Força de segurança
RSA	3072 bits	$\approx$ 128 bits
ECDSA	256 bits	$\approx$ 128 bits
Ed25519	256 bits	$\approx$ 128 bits

No trânsito, os motoristas precisam manter uma certa distância do veículo à frente, para que não ocorra colisão em caso de sinistro ou frenagem brusca. Esta distância, chamada distância segura, depende de diversos fatores, como a velocidade do veículo e as condições da pista, do condutor e do próprio veículo.

Para [Fambro et al. 2000], o tempo gasto para que um veículo pare após o acionamento dos freios, depende não só da velocidade, mas também, se o veículo está equipado com freios ABS (*Antiblockier-Bremssystem* - sistema de frenagem anti-derrapante), se a pista está seca ou molhada ou se a manobra é em uma curva, por exemplo. Em redes veiculares, quando uma mensagem é enviada por comunicação V2V, o tempo de transmissão da mensagem também interfere na distância segura [Li et al. 2020].

Segundo [Chen et al. 2013], a distância de parada é a soma das distâncias de reação, acúmulo de pressão e frenagem, onde: distância de reação é a distância percorrida pelo veículo desde o momento que o motorista observou o incidente até o momento em que o mesmo acionou os freios; distância de acúmulo de pressão é a distância que o

veículo se desloca durante o início da ação de frenagem até a configuração completa da pressão de frenagem; distância de frenagem é espaço que o veículo percorre do momento em que os freios foram acionados, até o momento em que o veículo para na via.

A Tabela 2, exibe todas as variáveis utilizadas nos cálculos de distância reação e distância de acúmulo de pressão.

Tabela 2. Variáveis utilizadas nos cálculos de distância de parada

Variável	Descrição	Unidade
V	velocidade	m/s
D_r	distância de reação	m
D_p	distância de acúmulo de pressão	m
t_r	tempo de reação	s
t_p	tempo de acúmulo de pressão	s

Para [Ruhai et al. 2010], o tempo de reação do motorista, depende de vários fatores, entre eles, a experiência do condutor, o sexo e condições físicas e psicológicas. Segundo [Chen et al. 2013], a distância de reação é dada pelo produto do tempo de reação do motorista, dado em segundos, e a velocidade do veículo em metros por segundo. O tempo de reação varia de $0,74$ a $1,17$ segundos. A Equação 1, exibe este cálculo.

$$D_r = t_r * V \tag{1}$$

Segundo [Chen et al. 2013], a distância de acúmulo de pressão é dada pelo produto do tempo de acúmulo de pressão, dado em segundos, e a velocidade do veículo em metros por segundo. Este tempo varia de $0,3$ e $0,75$ segundos. A Equação 2, mostra este cálculo.

$$D_p = t_p * V \tag{2}$$

A distância de frenagem leva em consideração, além da velocidade, as dimensões do veículo, a eficiência do freio, atrito da pista entre outras, como pode ser visto na Tabela 3. Seu cálculo dado pela Equação 3 [Wong 2008].

$$D_b = \frac{\gamma W}{g\rho A_h A_w C_d} \ln(1 + \frac{\frac{(\rho A_h A_w C_d)}{2} V^2}{\eta_b(\mu + f_r)W\cos\theta_s \pm W\sin\theta_s}) \tag{3}$$

A distância total de parada segura (D_s) é dada pela Equação 4, que é o somatório das distâncias de reação, acúmulo de pressão e frenagem.

$$D_s = D_r + D_p + D_r \tag{4}$$

3. Trabalhos Relacionados

Em [Wasef and Shen 2013], o `EMAP` (*Expedite Message Authentication Protocol*) é sugerido para VANET, com foco no processo de verificação da lista de certificados revogados. O `EMAP` usa o algoritmo `HMAC` (*Hash Message Authentication Code*) para autenticação de mensagens e um sistema probabilístico de alocação de chaves para a distribuição de

Tabela 3. Variáveis utilizadas nos cálculos de distância de frenagem

Variável	Descrição	Unidade
D_b	distância de frenagem	m
γ	fator de massa equivalente	$-$
W	massa total do veículo (passageiros, bagagens, etc)	kg
g	aceleração da gravidade	m/s^2
ρ	densidade do ar	kg/m^3
A_h	altura do veículo	m
A_w	largura do veículo	m
C_d	coeficiente de resistência aerodinâmica	$-$
V	velocidade do veículo	m/s
η_b	eficiência da frenagem	$-$
μ	coeficiente de adesão da pista	$-$
f_r	coeficiente de resistência do rolamento	$-$
θ_s	ângulo de inclinação horizontal da pista	^{o} (graus)
$\pm$	positivo para subida; negativo para descida	$-$

chaves entre veículos. Nos testes realizados, o protocolo reduziu significativamente a perda de mensagens em comparação com outros modelos. Porém, é necessário criar e distribuir um grande número de chaves entre os veículos, o que aumenta a complexidade do gerenciamento de chaves.

Devido às restrições de conectividade da VANET e para reduzir a sobrecarga inserida por `ECDSA`, os autores de [Sakhreliya and Pandya 2014] criaram um sistema híbrido que usa infraestrutura de chave pública com criptografia simétrica. Assim, enquanto um mecanismo de criptografia assimétrica pura tem um tempo de dois milissegundos para geração de mensagens e cinco milissegundos para a verificação do mesmo, usando o sistema proposto pelos autores, o tempo de processamento é reduzido para 26 microssegundos em cada uma das operações anteriores.

Em [Wang and Yao 2017], os autores apresentaram um modelo de autenticação em duas etapas: autenticação com criptografia assimétrica por meio de um certificado digital de longo prazo e criptografia simétrica com chave mestre trocada com a `RSU`. O trabalho também aborda a questão de se referir às listas de certificados revogados. Segundo os autores, as simulações realizadas atingem o objetivo proposto, embora crie uma sobrecarga para a verificação de certificados revogados.

Os autores [Kushwah et al. 2019] propuseram um método de autenticação de mensagens em redes veiculares usando o algoritmo `ECDSA`, a fim de garantir a autenticidade e não repúdio dos veículos. Com o algoritmo utilizado, os autores conseguem baixos tempos de assinatura e verificação de mensagens.

Pelo nosso conhecimento, até o momento não existem trabalhos em VANET, utilizando o algoritmo `Ed25519`, que é objeto de estudo de nosso trabalho.

4. Segurança na mensagem para comunicação veicular

Este artigo apresenta o fluxo da comunicação veicular considerando a simulação de um CA (*Certificate Authority*) e emulação de um dispositivo veicular. Além disso, são calcu-

ladas e comparadas as distâncias de parada para cada um dos três algoritmos de assinatura digital, considerando-se forças de segurança similares.

Computando a assinatura digital: para que os veículos possam assinar as mensagens enviadas, é necessário que eles possuam um certificado digital e um par de chaves pública e privada, emitidos por uma `CA` (*Certificate Authority*) válida. No início da simulação, cada veículo recebe da `CA` seu certificado e chaves. Para isso, foi desenvolvido em `C++`, o `CAServer`, que usa a biblioteca de criptografia `Crypo++ 8.2.0`, e permanece em execução em uma máquina virtual, distribuindo certificados e chaves para todos os veículos que entram na simulação. O `CAServer` permanece ouvindo na porta `1393/TCP`, aguardando por conexões do simulador (ver Seção 5) via *socket*. Para cada veículo que entra na simulação, o simulador abre um *socket* com o `CAServer`, enviando a ele um arquivo no formato `json` contendo suas informações e recebe de volta outro `json` com seu certificado digital e suas chaves.

Em uma situação real, os veículos receberiam os certificados e chaves de forma segura, indo até a agência de uma autoridade certificadora válida. Mas nas simulações, o certificado e chaves foram entregues aos veículos na inicialização. A fim de testar e comparar o desempenho dos algoritmos de assinatura digital de forma real e validar seu uso em uma rede veicular, foi utilizado um hardware embarcado para realizar este processo. O hardware utilizado foi um `Gumstix Overo FireSTORM-P`, com sistema operacional `Ubuntu Core 15.04` e a placa de expansão `Tobi`. A configuração detalhada do hardware pode ser vista na Tabela 4.

Tabela 4. Configuração do hardware utilizado

Componente	Descrição
CPU	DaVinci DM3730 ARM Cortex-A8, de 800MHz a 1GHz
Memória RAM	512MB DDR LPDRAM
Memória Flash	512MB NAND
Rede	10/100Mbps Ethernet

A escolha do `Gumstix` foi motivada pela disponibilidade do equipamento no laboratório e por ser um dispositivo de processamento limitado. Se os algoritmos testados podem ser executados neste dispositivo limitado, também podem ser executados em *hardwares* superiores e terão resultados proporcionalmente semelhantes.

A comunicação do simulador e o hardware, se dá através de um `socket TCP` que o simulador abre com o dispositivo toda vez que um veículo precisa assinar ou checar a assinatura de uma mensagem. O `OBUEmulator` foi desenvolvido em `C++`, utilizando a biblioteca de criptografia `Crypo++ 8.2.0`, e implementa os três algoritmos de assinatura digital: `RSA`, `ECDSA` e `Ed25519`. O `OBUEmulator` fica em execução no hardware, escutando conexões na porta `8817/TCP`. A Figura 3 ilustra o procedimento de comunicação entre o simulador `OMNet++` e o `OBUEmulator`. O `OBUEmulator` funciona como uma `OBU` compartilhada para todos os veículos da simulação, que recebe e envia strings no formato `json`, através de um *socket* `TCP`.

Quando um veículo precisa enviar uma mensagem, o simulador abre uma conexão com o `OBUEmulator` (localizado no hardware real) e envia um `json` contendo: o tipo

de algoritmo de assinatura digital que será utilizado; a mensagem para assinar e a chave privada. O hardware então processa o pedido e devolve para o simulador a assinatura da mensagem e o tempo gasto no processo.

Da mesma forma, quando um veículo recebe uma mensagem, o simulador abre uma conexão com o `OBUEmulator` enviando um `json` contendo: o tipo de algoritmo de assinatura digital; o certificado do veículo, juntamente com sua chave pública e a assinatura da CA sobre o certificado; a mensagem para checar a assinatura; a assinatura digital da mensagem e a chave pública da CA. O dispositivo valida as assinaturas da CA sobre o certificado do veículo (utilizando a chave pública da CA) e da mensagem (utilizando a chave pública do veículo). Após realizado este procedimento, o hardware envia para o simulador se as validações estão corretas e o tempo gasto nas operações.

Em ambos os procedimentos, o simulador `OMNET++` adiciona ao tempo de simulação, o tempo recebido pelo `OBUEmulator`, a fim de atrasar a comunicação, com o *overhead* do tempo gasto no processamento.

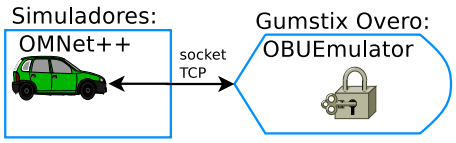

Figura 3. Comunicação entre os simuladores e o hardware

Para que houvesse justiça nos testes, foi utilizado em todos os algoritmos testados uma força de segurança similar (nível de segurança em bits). Os algoritmos foram implementados com os tamanhos de chaves de acordo com a Tabela 5. Na qual, segundo [Kumar 2006] e [Bernstein et al. 2012], uma chave de `256 bits ECC`, equivale a uma chave de aproximadamente `3072 bits RSA` e tem um nível de segurança de `128 bits`.

Tabela 5. Tamanho das chaves dos algoritmos

Algoritmo	Tamanho da chave
`RSA`	3072 bits
`ECDSA`	256 bits
`Ed25519`	256 bits

Computando a distância de parada: além dos tempos de assinatura digital e verificação de assinatura, foram avaliados também os tempos de reação do motorista e a distância total de parada. As distâncias de reação e acúmulo de pressão foram calculados utilizando o pior caso (Tabela 6) , com tempo de reação igual a $1,17$ segundos e tempo de acúmulo de pressão igual a $0,75$ segundos. A Equação 5 mostra o tempo total de reação e acumulo de pressão, onde, V é a velocidade do veículo e D_{rp} é o tempo de reação somado ao tempo de acúmulo de pressão.

$$D_{rp} = (1,17 * V) + (0,75 * V) => D_{rp} = 1,92 * V \quad (5)$$

Para os cálculos de distância de frenagem da Equação 3, foram utilizados valores de um veículo de passeio, de acordo com a Tabela 6, baseados nos valores utilizados

por [Chen et al. 2013], no qual, presume-se o seguinte cenário: i) veículo de passeio com tração frontal e sem freios ABS; ii) quatro passageiros a bordo do veículo; iii) temperatura do ar de $25°$, ao nível do mar; iv) estrada de asfalto molhado, com $0°$ de inclinação (reta).

Tabela 6. Valores utilizados nos cálculos de distância de parada

Variável	Descrição	Valor
γ	fator de massa equivalente	$1,04$
W	massa total do veículo (passageiros, bagagens, etc)	1440
g	aceleração da gravidade	$9,8$
ρ	densidade do ar	$1,18$
A_h	altura do veículo	$1,41$
A_w	largura do veículo	$1,55$
C_d	coeficiente de resistência aerodinâmica	$0,45$
V	velocidade do veículo	$-$
η_b	eficiência da frenagem	$0,8$
μ	coeficiente de adesão da pista	$0,45$
f_r	coeficiente de resistência do rolamento	$0,015$
θ_s	ângulo de inclinação horizontal da pista	0

Com os dados da Tabela 6, a Equação 3, pode ser simplificada, e substituída neste cenário, pela Equação 6.

$$D_b = 131,72 \ln[1 + (1,083 * 10^{-3} V^2)] \tag{6}$$

Portanto, para este cenário, o resultado da distância total de parada segura é o somatório das distâncias de reação do motorista, acúmulo de pressão e frenagem, conforme mostra a Equação 7.

$$D_s = 131,72 \ln[1 + (1,083 * 10^{-3} V^2)] + 1,92V \tag{7}$$

Algoritmos de geração, assinatura e validação de mensagens: o Algoritmo 1 exemplifica o processo de geração e assinatura da mensagem, no qual é escolhido o algoritmo de assinatura digital (linha 1) logo no início da simulação. Quando ocorre um acidente (linha 2), uma mensagem de acidente é criada (linha 3) e uma conexão com o `OBUEmulator` é feita (linha 4), enviando: a mensagem que será assinada (`msg`), o algoritmo utilizado (`alg`) e a informação, se é uma operação de assinatura ou validação de assinatura, neste caso, assinatura (`SIGN`). O `OBUEmulator` retorna a assinatura (`ass`) da mensagem e o tempo gasto na operação (`tempoAss`). Em seguida, aguarda-se um tempo de `tempoAss` segundos (linha 5) e envia a mensagem com a assinatura e a chave pública do remetente (`msg+ass+chavePub`) via *broadcast* (linha 6).

O Algoritmo 2, mostra o processo de validação, ou checagem, da assinatura de uma mensagem recebida. No início da simulação é escolhido o algoritmo de assinatura digital (linha 1). Caso uma mensagem seja recebida (linha 2), uma conexão com o `OBUEmulator` é estabelecida (linha 3), enviando a mensagem com a assinatura digital e a chave pública do remetente, o algoritmo utilizado e a sinalização de operação, indicando uma validação de assinatura (`msg+ass+chavePub+alg+CHECK`). O `OBUEmulator` responde com o resultado da validação (`resultado`) e o tempo gasto na operação

Algorithm 1 Veículos que se acidentam (enviam mensagens)

```
1: alg = AlgAssinaturaDigital
2: if ocorreuAcidente then
3:     msg = criaMensagem()
4:     [ass, tempoAss] = conectaOBUSim(msg, alg, SIGN)
5:     aguarda(tempoAss)
6:     enviaMsg(msg+ass+chavePub)
7: end if
```

(`tempoGasto`). Caso o resultado da validação seja positivo (linha 4), pega-se a velocidade atual do veículo (`vel`, linha 5) e calcula-se a distância total de parada, com a soma usando a Equação 7 (linha 6) e soma-se todos os tempos.

Algorithm 2 Veículos que recebem mensagens de acidente

```
1: alg = AlgAssinaturaDigital
2: if msgRecebida then
3:     [resultado, tempoGasto] = conect2OBUSim(msg+ass+chavePub, alg, CHECK)
4:     if resultado then
5:         vel = getVelocidade()
6:         distanciaTotalParada = timeCheck*vel + calcDistanciaParada(vel)
7:     else
8:         abort()
9:     end if
10: end if
```

5. Resultados e Discussões

Os experimentos foram realizados utilizando o *framework opensource* `Veins 5` [Sommer et al. 2011], que implementa os protocolos de comunicação de rede veicular do padrão WAVE (*Wireless Access in Vehicular Environments*). O *framework* utiliza o simulador de tráfego urbano `SUMO 1.2` [Lopez et al. 2018], para simular o tráfego de veículos em uma via e o simulador de eventos discretos `OMNet++ 5.5` [Varga 2010], para simular a comunicação de rede entre os veículos. Cada algoritmo de assinatura digital (`RSA`, `ECDSA` e `Ed255190`) foi executado 33 vezes, totalizando 99 simulações.

Quanto ao tempo de processamento dos algoritmos testados (assinatura e verificação), o algoritmo `Ed25519` apresentou o melhor resultado em relação aos demais, sendo cerca de 9 vezes mais rápido que o `ECDSA` e 42 vezes mais rápido que o `RSA`. A Fig. 4 mostra o gráfico desta comparação, usando um nível de confiança de 99%. Como os valores encontrados são muito pequenos, a Tabela 7 mostra os tempos médios (em milissegundos) e o intervalo de confiança (IC) para cada algoritmo testado.

A Tabela 8, apresenta os dados em relação a distância total de parada do veículo a 110km/h, usando um nível de confiança de 99% e que leva em consideração as distâncias de: assinatura da mensagem, verificação da mensagem, reação do motorista, acúmulo de pressão dos freios e frenagem. Usando o algoritmo `Ed25519`, o veículo percorre em média, uma distância total de 151 metros antes de parar. Com o `ECDSA`, o veículo percorre, em média, cerca de 3 metros a mais que o `Ed25519`. Já o `RSA`, percorre uma distância de $165,6$ metros, 14 metros a mais que o algoritmo com melhor resultado.

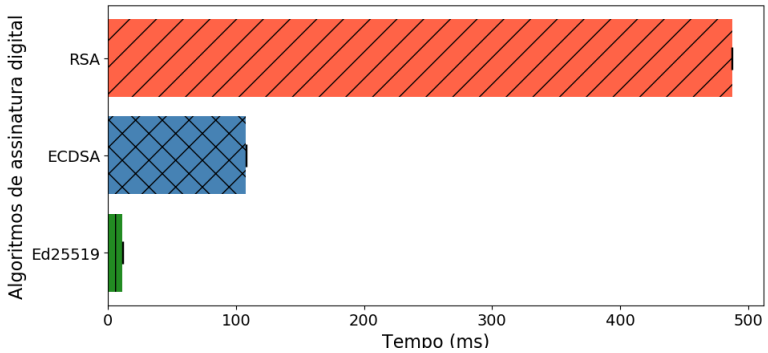

Figura 4. Tempo de processamento dos algoritmos de assinatura digital

Estas distâncias são para veículos a 110km/h e configurações de acordo com a Tabela 3. Quanto mais rápido estiver o veículo, maior será a diferença entre um algoritmo e outro, proporcionalmente, já que a distância é o produto entre a velocidade e o tempo gasto no processamento da assinatura e verificação.

Nos experimentos, fica evidente que o uso do `RSA`, com o nível de segurança utilizado, não é viável em redes veiculares, devido às restrições de tempo. Os algoritmos `ECDSA` e `Ed25519` tiveram valores próximos, com uma diferença de distância de parada de três metros. Mas em se tratando de redes veiculares, este valor pode ser suficiente para evitar um acidente. É notório que o algoritmo de assinatura digital `Ed25519` é o mais indicado para uso em VANET, pois oferece o mesmo nível de segurança em bits que os demais e com melhor desempenho, mesmo rodando em um hardware embarcado e com restrições de processamento.

Tabela 7. Tempo de processamento dos algoritmos

Algoritmo	Média (ms)	IC (99%)
`RSA`	$487,41$	$\pm 0,48$
`ECDSA`	$107,78$	$\pm 0,37$
`Ed25519`	$11,44$	$\pm 0,18$

Tabela 8. Distância total de parada dos algoritmos

Algoritmo	Média (m)	IC (99%)
`RSA`	$165,62$	$\pm 0,015$
`ECDSA`	$154,02$	$\pm 0,010$
`Ed25519`	$151,08$	$\pm 0,005$

6. Conclusões e Trabalhos Futuros

O uso de assinatura digital em mensagens críticas de trânsito é fundamental para garantir a integridade e o propósito da rede veicular. Este trabalho avaliou três algoritmos de assinatura digital em VANET (`RSA`, `ECDSA` e `Ed25519`) e apontou o `Ed25519` como sendo o mais eficiente, mesmo em um ambiente tão dinâmico como em redes veiculares, usando um hardware embarcando, com restrições de computação. Pelos experimentos, sugere-se que este algoritmo seja padrão para uso em redes veiculares.

Como trabalhos futuros, pretende-se testar os mesmos algoritmos de assinatura digital, em um ambiente com alta densidade de veículos. Pretende-se ainda investigar o uso do `Ed25519` em um ambiente real, avaliando outras métricas.

Atualmente está em desenvolvimento um *framework* utilizando assinatura digital com o algoritmo `Ed25519` e um sistema de reputação que trata vários tipos de ataques em

redes veiculares, por meio de uma arquitetura que contempla: o uso de CA para emissão de certificados digitais e calculo de reputação; RSUs para a distribuição dos certificados gerados pela CA e captação de *feed-back* de veículos.

AGRADECIMENTOS

Os autores agradecem ao apoio financeiro da agência CAPES, FAPEMIG e CNPq.

Referências

Barbara, C. (2018). Digital Signatures, pages 1093–1099. Springer New York, New York.

Barker, E. (2016). Nist special publication 800-57 – part 1 revision 4 recommendation for key management. Technical report, NIST National Institute of Standards and Technology, 100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930 USA.

Bernstein, D. J. (2006). Curve25519: New Diffie-Hellman Speed Records. In Yung, M., Dodis, Y., Kiayias, A., and Malkin, T., editors, Public Key Cryptography - PKC 2006, pages 207–228, Berlin, Heidelberg. Springer Berlin Heidelberg.

Bernstein, D. J., Duif, N., Lange, T., Schwabe, P., and Yang, B.-Y. (2012). High-speed high-security signatures. Journal of Cryptographic Engineering, 2(2):77–89.

Bernstein, D. J., Josefsson, S., Lange, T., Schwabe, P., and Yang, B.-Y. (2015). Eddsa for more curves. IACR Cryptology ePrint Archive, 2015:677.

Bhover, S. U., Tugashetti, A., and Rashinkar, P. (2017). V2X communication protocol in VANET for co-operative intelligent transportation system. In International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), pages 602–607.

Chen, Y., Shen, K., and Wang, S. (2013). Forward collision warning system considering both time-to-collision and safety braking distance. In 2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA), pages 972–977.

de Bettio, R. W., Correia, L. H. A., and Moraes, H. P. (2019). Redes Veiculares - Registro de software INPI BR512019002553-9 - Dezembro 2019.

Fambro, D. B., Koppa, R. J., Picha, D. L., and Fitzpatrick, K. (2000). Driver braking performance in stopping sight distance situations. Transportation Research Record, 1701(1):9–16. doi:10.3141/1701-02.

Hartenstein, H. and Laberteaux, L. P. (2008). A tutorial survey on vehicular ad hoc networks. IEEE Communications Magazine, 46(6):164–171.

IANIX (2019). Things that use ed25519. https://ianix.com/pub/ed25519-deployment.html.

Johnson, D., Menezes, A., and Vanstone, S. (2001). The elliptic curve digital signature algorithm (ecdsa). International Journal of Information Security, 1(1):36–63.

Josefsson, S. and Liusvaara, I. (2017). Edwards-Curve Digital Signature Algorithm (EdDSA). RFC 8032, RFC Editor.

Kalra, S. and Sood, S. K. (2011). Elliptic curve cryptography: Survey and its security applications. In Proceedings of the International Conference on Advances in Computing and Artificial Intelligence, ACAI '11, pages 102–106, New York, NY, USA. ACM.

Kumar, S. S. (2006). Elliptic Curve Cryptography for Constrained Devices. PhD thesis, Ruhr-University Bochum.

Kushwah, R., Kulshreshtha, A., Singh, K., and Sharma, S. (2019). Ecdsa for data origin authentication and vehicle security in vanet. In 2019 Twelfth International Conference on Contemporary Computing (IC3), pages 1–5. 8844912.

Li, J., Zhang, Y., Shi, M., Liu, Q., and Chen, Y. (2020). Collision avoidance strategy supported by lte-v-based vehicle automation and communication systems for car following. Tsinghua Science and Technology, 25(1):127–139. 8768212.

Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., and Wießner, E. (2018). Microscopic Traffic Simulation using SUMO. In XXI IEEE Int. Conf. on Intelligent Transportation Systems.

Manvi, S. S., Kakkasageri, M. S., and Adiga, D. G. (2009). Message authentication in vehicular ad hoc networks: Ecdsa based approach. In 2009 International Conference on Future Computer and Communication, pages 16–20. 5189734.

Mishra, R., Singh, A., and Kumar, R. (2016). Vanet security: Issues, challenges and solutions. In 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), pages 1050–1055.

Perbawa, M. R., Afryansyah, D. I., and Sari, R. F. (2017). Comparison of ecdsa and rsa signature scheme on nlsr performance. In 2017 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob), pages 7–11. 8284007.

Ravi, K. and Kulkarni, S. A. (2013). A secure message authentication scheme for vanet using ecdsa. In 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), pages 1–6. 6726769.

Rivest, R. L., Shamir, A., and Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM, 21(2):120–126. Rivest:1978:MOD:359340.359342.

Ruhai, G., Weiwei, Z., and Zhong, W. (2010). Research on the driver reaction time of safety distance model on highway based on fuzzy mathematics. In 2010 International Conference on Optoelectronics and Image Processing, volume 2, pages 293–296. 5663465.

Sakhreliya, S. C. and Pandya, N. H. (2014). Pki-sc: Public key infrastructure using symmetric key cryptography for authentication in vanets. In 2014 IEEE International Conference on Computational Intelligence and Computing Research, pages 1–6. 7238326.

Sommer, C., German, R., and Dressler, F. (2011). Bidirectionally coupled network and road traffic simulation for improved ivc analysis. IEEE Transactions on Mobile Computing, 10(1):3–15. 5510240.

Varga, A. (2010). OMNeT++, pages 35–59. Springer Berlin Heidelberg, Berlin, Heidelberg. Varga2010.

Wang, S. and Yao, N. (2017). Liap: A local identity-based anonymous message authentication protocol in vanets. Computer Communications, 112:154 – 164. WANG2017154.

Wasef, A. and Shen, X. (2013). Emap: Expedite message authentication protocol for vehicular ad hoc networks. IEEE Transactions on Mobile Computing, 12(1):78–89. 6081877.

Wong, J. Y. (2008). Theory of Ground Vehicles. Wiley, 4 edition.

World Heath Organization (2019). Road traffic injuries. `https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries`.