Avaliação e Comparação de Desempenho entre Algoritmos de Criptografia de Curva Elíptica com El-Gamal baseados nas Bibliotecas MIRACL e RELIC

Daniel Fernando Pigatto, Natássya Barlate Floro da Silva,
Kalinka Regina Lucas Jaquie Castelo Branco
ICMC – USP
{pigatto,kalinka}@icmc.usp.br, ays@grad.icmc.usp.br

***Resumo*—Este trabalho apresenta uma comparação entre duas implementações de um mesmo algoritmo de criptografia de curva elíptica (ECC) baseado em El-Gamal. Cada implementação apoia-se em diferentes bibliotecas criptográficas, sendo elas MIRACL e RELIC. A aplicação deste algoritmo vai ao encontro das necessidades mais comuns a sistemas embarcados críticos, com foco em sistemas aéreos e terrestres não-tripulados (VANTs e VTNTs), buscando contemplar as particularidades destes cenários, tais como limitações de recursos computacionais e de suprimento de energia. Um dos desafios a serem contemplados é o estudo e a avaliação do impacto do uso de cada biblioteca por meio de uma avaliação de desempenho baseada em técnicas estatísticas. Os resultados mostram a melhor solução e a influência exercida por cada algoritmo nos tempos de resposta de cada experimento realizado.**

I. Introdução

O uso de sistemas embarcados é cada vez mais frequente em ambientes domésticos, empresariais e para monitoramento de fenômenos naturais. Uma outra classe de sistemas embarcados, os de ordem crítica, engloba sistemas de monitoramento do meio ambiente, operações militares e aplicações agrícolas, entre outras. Estes sistemas exigem um cuidado maior em relação aos dados coletados, trafegados entre dois sistemas ou com bases de controle, ou ainda com relação à pilotagem de veículos aéreos e terrestres considerados autônomos e não-tripulados.

Para isso, a segurança atua com o objetivo de prover uma abordagem adequada para cada cenário especificamente, visando assegurar o sistema contra entidades maliciosas que possam deliberadamente obter acesso a informações ou modificar o funcionamento destes dispositivos, e levando em consideração limitações de recursos apresentadas pelos mesmos.

O artigo segue organizado em seções. A seção II apresenta uma revisão bibliográfica sobre sistemas embarcados críticos. A seção III dá continuidade à revisão bibliográfica com foco em segurança. Na seção IV, a revisão bibliográfica contempla segurança especificamente para sistemas embarcados críticos. Experimentos e resultados são apresentados na seção V. E a seção VI mostra algumas conclusões.

II. Sistemas Embarcados Críticos

Dispositivos responsáveis por funções dedicadas, como os presentes em um micro-ondas, por exemplo, são chamados de sistemas embarcados. Diversas são as definições de sistemas embarcados encontradas na literatura. Netrino [5] define sistema embarcado como uma combinação de hardware e software, e talvez mais algumas peças mecânicas, destinadas a executar uma função específica. Em alguns casos, são parte de um sistema ou produto.

Estes sistemas são atualmente encontrados em muitas situações do dia a dia. Vahid & Givargis enumeram algumas aplicações, entre elas: eletrônicos portáteis (telefones celulares, *pagers*, câmeras digitais, calculadoras e PDAs), eletrodomésticos (fornos micro-ondas, termostatos, sistemas de segurança, máquinas de lavar e sistemas de iluminação), automação para escritório (faxes, impressoras e *scanners*), equipamentos de negócio (caixas registradoras, sistemas de alarmes e leitores de cartão) e automóveis (injeção eletrônica, freios antitravamento e suspensão ativa) [10].

A. Características

A computação embarcada difere da computação tradicional em vários aspectos. A busca por funcionalidade é comum entre ambas, porém ao se trabalhar com sistemas embarcados existem outros fatores que devem ser levados em consideração tais como limitações de recursos de processamento, memória e energia.

Programadores de propósito geral não têm, na maioria dos casos, preocupações com o desempenho de suas aplicações, uma vez que podem utilizar recursos disponíveis sem limitações e precisam que seus programas rodem "rápido o suficiente", e não "o mais rápido possível".

Em sistemas embarcados, por outro lado, desempenho é um objetivo claro de todo desenvolvedor, ou seja, os programas devem atender a prazos bem definidos. No cerne da computação embarcada está a computação de tempo real, a qual é a ciência e a arte de programar respeitando prazos específicos de tempo [11]. O programa recebe dados de entrada e possui um prazo para realização das operações computacionais necessárias. Se o programa não produzir o resultado necessário dentro do prazo, então o programa não funciona, mesmo que a eventual saída produzida seja correta.

Exemplos de sistemas embarcados críticos são os veículos terrestres não-tripulados, dispositivos utilizados principalmente em aplicações militares, ambientais e agrícolas para automatizar processos.

III. Segurança

Segundo Stapko [9], segurança de computadores consiste em proteger informações pessoais ou confidenciais e/ou recursos computacionais de indivíduos ou organizações que poderiam deliberadamente destruir ou utilizar tais informações para fins maliciosos. Algumas propriedades devem ser garantidas para uma completa e

eficaz implementação de segurança em sistemas computacionais [1][3][8]:

- **Confidencialidade**: Garantia de que somente o remetente e o destinatário pretendido terão o poder de entender o conteúdo da mensagem. Se algum intruso conseguir interceptar a mensagem, não deverá conseguir extrair informações do texto cifrado (disfarçado ou ilegível).
- **Autenticidade**: Garantia de que a entidade participante da comunicação é realmente quem ela afirma ser. Remetente e destinatário precisam confirmar a identidade da outra parte envolvida.
- **Integridade e não-repúdio de mensagem**: Integridade se refere à confiabilidade dos dados ou recursos, ou seja, trata-se da garantia de que não houve mudanças impróprias ou não autorizadas durante a comunicação (modificação, inserção, exclusão ou repetição). O receptor pode, ainda, comprovar que a mensagem veio de um remetente específico. Trata-se de uma proteção de negação, por parte de uma das entidades envolvidas na comunicação, de ter participado de parte ou de toda a comunicação, propriedade conhecida como não-repúdio.
- **Disponibilidade**: Refere-se à capacidade de acesso a informações e serviços sempre que necessário, ou seja, um sistema estará disponível se oferecer os serviços de acordo com o projeto do sistema sempre que os usuários os solicitarem.

IV. Segurança de Sistemas Embarcados Críticos

A aplicação de algoritmos criptográficos em sistemas computacionais acrescenta uma camada adicional de execução que, consequentemente, aumenta o tempo de resposta durante a execução de tarefas. Segurança e desempenho são conceitos antagônicos, isto é, geralmente, quando aumenta-se o nível de segurança de uma solução criptográfica, perde-se desempenho e vice-versa. Isto implica na necessidade da busca por um equilíbrio entre desempenho e segurança de acordo com a aplicação final.

Quando trabalha-se com veículos autônomos de ordem crítica, o papel da segurança é fundamental para garantir que estes dispositivos possam funcionar corretamente sem sofrer intervenções de entidades maliciosas que possam desviá-los de suas rotas e/ou áreas de atuação pré-definidas. Para tal, leva-se em consideração alguns elementos comuns a cenários onde normalmente emprega-se este tipo de veículo para propor uma solução de segurança que garanta o bom funcionamento de tais dispositivos.

Um fator importante é a delimitação de uma área na qual o veículo deverá atuar, podendo deslocar-se livremente dentro dos limites pré-definidos. Um sistema deve monitorar a posição do veículo por meio de suas coordenadas de localização e a área imposta para sua atuação. Quando o veículo ultrapassar os limites e não retornar em um determinado prazo de tempo muito curto, comandos de retorno devem ser enviados para que evite-se a perda de informações ou o comprometimento do mesmo caso depare-se com obstáculos não previstos.

Esta comunicação deve fazer uso de mecanismos de autenticação para garantir que os comandos enviados sejam provenientes de uma base de controle autorizada a realizar aquela ação (ou mesmo de um controlador responsável que esteja monitorando as ações do veículo). Esta é uma forma de impedir que entidades externas deliberadamente disparem comandos para mudança de direção do dispositivo para benefício próprio ou para provocar danos aos dispositivos em questão.

Além desta autenticação, existe a necessidade, em alguns casos, de se assegurar a confidencialidade e a integridade de informações enviadas em tempo real para uma base de controle ou para outros veículos envolvidos em uma determinada operação. Isto implica na necessidade do uso de algoritmos simétricos e assimétricos para troca segura dessas informações.

Um algoritmo foco de pesquisas atualmente em sistemas embarcados críticos é o ECC (*Elliptic Curve Cryptography*). Ele trabalha com chaves de tamanho consideravelmente menores e, consequentemente, executa cálculos computacionalmente menos pesados. Apesar disso, o nível de segurança oferecido pelo algoritmo com chaves pequenas é equivalente ao emprego de outros algoritmos de chave pública com chaves de tamanhos maiores. O nível de segurança de uma implementação de curvas elípticas com chave de 160 bits é equivalente ao RSA com chave de tamanho 1024 bits [4].

V. Experimentos e Resultados

O foco de aplicação do algoritmo ECC é justamente na transmissão de informações entre dispositivos embarcados críticos, tanto no cenário aéreo (VANTs – Veículos Aéreos Não-Tripulados), quanto no terrestre (VTNTs – Veículos Terrestres Não-Tripulados), contemplando as particularidades de cada cenário e levando em consideração as preocupações inerentes a este tipo de sistema.

Dois algoritmos foram desenvolvidos e analisados comparativamente em aspectos relativos à criptografia de arquivos de imagem. O método escolhido para implementação foi o de El-Gamal sobre ECC, combinando as propriedades de curvas elípticas com o método de criptografia das trocas de mensagens de El-Gamal. O seu funcionamento se dá da seguinte forma: os dois usuários devem compartilhar a mesma curva elíptica e um determinado ponto P. Então, cada um deve escolher um número aleatório a, que funciona como sua chave privada, e multiplicar o ponto conhecido por ele, obtendo aP, que se torna a sua chave pública. A chave pública é transmitida para o outro usuário no início da comunicação, que possui como chave pública bP e, consequentemente, a chave privada b. Então, para a troca de mensagens o usuário precisa multiplicar a sua chave privada pela chave pública do outro usuário obtendo $b(aP)$ e somar este resultado na mensagem codificada em um número M. Portanto, a mensagem transmitida será $M + b(aP)$. Quando a mensagem for entregue ao receptor, ele será capaz de decodifica-la multiplicando sua chave privada pela chave pública do outro usuário, $(a(bP))$, e subtraindo do conteúdo total, $M + b(aP) - a(bP) = M$.

Na implementação utilizou-se duas bibliotecas como apoio para a realização de operações matemáticas: MIRACL (*Multiprecision Integer and Rational Arithmetic C/C++ Library*) devido a sua aplicação em [6], e RELIC 0.2.3 (*Efficient Library for Cryptography*) [7]. A MIRACL é uma biblioteca proprietária, mas livre para uso

educacional, produzida pela Shamus Software. Ela tem como objetivo se portar como uma ferramenta para os desenvolvedores de sistemas de criptografia e oferece as operações necessárias para lidar com números grandes, além de oferecer um suporte completo a curvas elípticas. A RELIC, por sua vez, é uma biblioteca desenvolvida por pesquisadores da UNICAMP, com o objetivo de oferecer ferramentas criptográficas baseadas em flexibilidade e eficiência. Ela oferece implementações de aritmética de números inteiros grandes, aritmética de campos binários e primos, curvas elípticas sobre campos primos, entre outras.

O algoritmo desenvolvido com a biblioteca MIRACL opera sobre blocos de mensagens com tamanho fixo de 18 caracteres e o algoritmo que utiliza a biblioteca RELIC opera sobre blocos de mensagens com tamanho fixo de 32 caracteres. Os algoritmos possuem fixos os parâmetros que definem a curva elíptica e o ponto utilizado em comum pelos usuários. Estas definições foram estabelecidas de acordo com alguns experimentos realizados que comprovaram melhor desempenho com os respectivos tamanhos de bloco de mensagem informados.

Os experimentos foram configurados de acordo com regras de avaliação de desempenho de sistemas computacionais [2]. Existe uma variedade de termos empregados na etapa de projeto e análise de experimentos, tais como variável de resposta, fatores, níveis e interações. Variável de resposta representa o resultado (saída) de um experimento. Normalmente, a variável de resposta é a medida de desempenho do sistema. Fatores são as variáveis que afetam a variável de resposta do sistema. Níveis são os valores que um determinado fator pode assumir. E a interação indica a dependência entre os fatores avaliados [2].

Os primeiros passos foram as definições de uma variável de resposta a ser avaliada, a quantidade de replicações necessárias e o ambiente de testes. O ambiente de testes utilizado foi um computador Pentium Dual-Core CPU T4300 2,10 GHz com 2 GB de memória RAM e o sistema operacional Linux Ubuntu 11.04. Para avaliar a eficiência da criptografia e comparar os resultados dos dois algoritmos desenvolvidos, a variável de resposta selecionada foi o tempo médio de resposta. O processo total realizado nos experimentos trata-se da criptografia de cada um dos arquivos de entrada. Cada experimento foi repetido 15 vezes, garantindo assim uma validação estatística visto que não houve um desvio padrão elevado entre os resultados obtidos.

Existem alguns modos de se realizar o planejamento de experimentos e o utilizado neste trabalho é o planejamento fatorial completo [2]. Neste tipo de planejamento são utilizadas todas as combinações possíveis considerando todos os fatores e níveis. Assim, é possível avaliar todos os fatores, determinar o efeito de cada fator nos experimentos e verificar as interações entre eles. A Tabela I apresenta as combinações possíveis de experimentos. O primeiro fator é a biblioteca utilizada, que possui dois níveis: MIRACL e RELIC. E o segundo fator é o tamanho da mensagem, que pode variar entre 181 KB e 319 KB, escolhidos aleatoriamente.

A Figura 1 mostra o gráfico de comparação entre os tempos médios de resposta obtidos por cada um dos algoritmos executados sobre o primeiro tamanho de mensagem (menor). O algoritmo baseado na biblioteca MIRACL tem um tempo consideravelmente superior em relação ao baseado em RELIC. Os tempos obtidos são de aproximadamente 37 e 23 segundos, respectivamente.

Tabela I. Configurações de experimentos.

Núm. Exp.	Biblioteca	Tamanho de mensagem
1	MIRACL	Tamanho 1 (181 KB)
2	MIRACL	Tamanho 2 (319 KB)
3	RELIC	Tamanho 1 (181 KB)
4	RELIC	Tamanho 2 (319 KB)

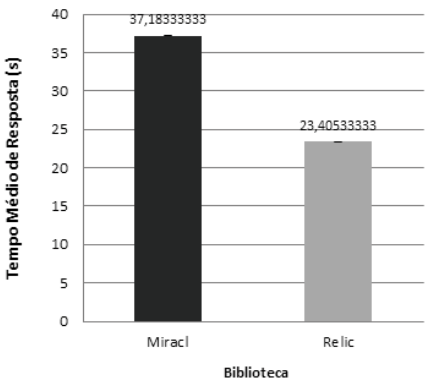

Figura 1. Comparação entre as bibliotecas MIRACL e RELIC com o primeiro tamanho de mensagem (181 KB).

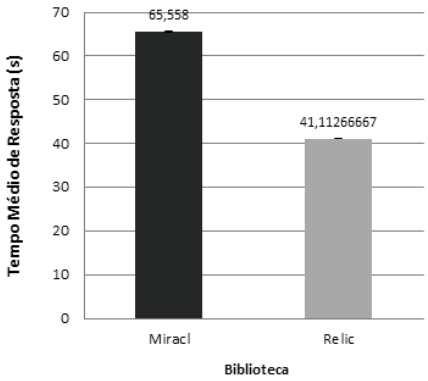

Figura 2. Comparação entre as bibliotecas MIRACL e RELIC com o segundo tamanho de mensagem (319 KB).

A Figura 2 apresenta o segundo gráfico de comparação, que mostra o desempenho dos algoritmos durante a criptografia do segundo tamanho de mensagem (maior). Houve uma elevação natural no tempo de resposta devido ao aumento de dados a serem processados, porém mantendo as mesmas características da comparação anterior. O algoritmo baseado em MIRACL teve um

tempo médio de resposta aproximado de 66 segundos, enquanto que o baseado em RELIC levou 41 segundos, aproximadamente, para realizar a operação completa.

Os gráficos apresentados sugerem um melhor desempenho para o algoritmo baseado na biblioteca RELIC em ambos os casos (Figura 1 e Figura 2). Para entender melhor os resultados apresentados, calculou-se o percentual de influência de cada fator da avaliação de desempenho, bem como a influência dos dois fatores associados sobre a variável de resposta selecionada. O gráfico apresentado na Figura 3 mostra que o fator A (algoritmo) teve uma influência de 40% nos resultados obtidos, o que é relevante para a comparação apresentada. O fator B (tamanho de mensagem), entretanto, exerceu uma influência maior sobre a variável de resposta, com um total de 57%. E por fim, os fatores AB associados tiveram baixa influência.

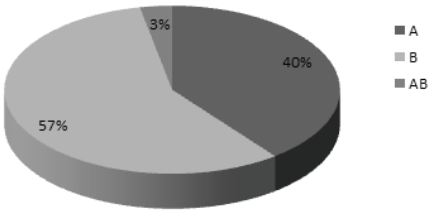

Figura 3. Influências de cada fator no tempo de resposta (A – algoritmo; B – Tamanho de mensagem; AB – Fatores associados).

Estes resultados mostram que o tamanho de mensagem influencia no resultado final da aplicação devido à diferença de quantidade de dados a serem criptografados. Porém, o objetivo deste artigo foi mostrar que a influência do algoritmo empregado é bastante considerável e, sendo o tempo de resposta um elemento crucial para sistemas embarcados críticos que frequentemente trabalham com tarefas de tempo real, o uso da biblioteca RELIC é mais indicado para implementação do algoritmo ECC, levando em consideração as condições do ambiente utilizado para estes experimentos, podendo haver variação quando os mesmos forem realizados em um ambiente com características semelhantes às de um sistemas embarcado de ordem crítica.

É importante ressaltar ainda que, num primeiro momento, os tempos apresentados podem ser classificados como altos, porém, considerando-se os tamanhos de chave e de imagem utilizados relativamente grandes e assumindo que sistemas embarcados críticos necessitam da aplicação de criptografia, na maioria dos casos, para envio de pequenos comandos, como por exemplo, de mudança de direção ou de alteração de rota, o desempenho apresentado vai ao encontro das expectativas, obtendo tempos de resposta muito baixos e sendo esta uma solução adequada para sistemas de tempo real, foco deste trabalho.

VI. Conclusões

O campo de aplicação de sistemas embarcados críticos exige o emprego de mecanismos que ofereçam níveis elevados de segurança, na maioria dos casos, e que garantam o funcionamento normal dos dispositivos, não comprometendo o resultado final dos mesmos. Para isso, torna-se necessário buscar soluções em criptografia que sejam computacionalmente eficazes e, ao mesmo tempo, seguras.

O estado da arte em sistemas embarcados críticos mostra algumas apostas no uso do algoritmo de curvas elípticas no escopo destes sistemas, devido às suas vantagens em relação a outros algoritmos assimétricos. Este trabalho mostrou uma comparação entre duas implementações do algoritmo baseadas em bibliotecas encontradas na literatura. Nestas condições, a biblioteca MIRACL obteve um desempenho inferior à biblioteca RELIC. Uma avaliação de desempenho foi apresentada com o objetivo de efetuar uma comparação entre as bibliotecas e apresentar a parcela de influência de cada fator envolvido nos experimentos realizados.

Como trabalho futuro existe a necessidade da execução destes algoritmos em hardware com limitações que assemelhem-se a sistemas embarcados. Além disso, no cenário de veículos aéreos e terrestres não-tripulados, estes algoritmos devem ser executados e submetidos a testes em movimento. Grande parte dos trabalhos encontrados na literatura prende-se a avaliações de desempenho onde os elementos da comunicação encontram-se fixos.

Referências

[1] Bishop, Matt. Introduction to Computer Security. San Francisco: Elsevier, 2004.

[2] Jain, R. The Art of Computer Systems Performance Analysis. JohnWiley & Sons, INC, 1991.

[3] Kurose, James F.; Ross, Keith W. Redes de Computadores e a Internet. 3. ed. São Paulo: Pearson Addison Wesley, 2006.

[4] Lenstra, Arjen K.; Verheul, Eric R. Selecting Cryptographic Key Sizes. Journal of Cryptology, 2001.

[5] Netrino. Embedded Systems Glossary. Disponível em: http://www.netrino.com/Embedded-Systems/Glossary-E Acesso em: Fev 2011.

[6] Ramachandran, Archana; Zhibin Zhou; Dijiang Huang. Computing Cryptographic Algorithms in Portable and Embedded Devices. IEEE International Conference on Portable Information Devices, 2007. PORTABLE07.

[7] RELIC. Relic Toolkit – Efficient Library for Cryptography. Disponível em: http://code.google.com/p/relic-toolkit/

[8] Stallings, William. Criptografia e Segurança de Redes: Princípios e Práticas. 4 ed. São Paulo: Pearson Prentice Hall, 2008.

[9] Stapko, Timothy. Practical Embedded Security. Burlington: Elsevier, 2008.

[10] Vahid, Frank; Givargis, Tony. Embedded System Design: A Unified Hardware/Software Introduction. United States of America: John Wiley & Sons, 2002.

[11] Wolf, Wayne. Computers as Components: Principles of Embedded Computing System Design. Burlington, USA: Morgan Kaufmann Publishers, 2008.