Aceleração de Assinaturas Baseadas em Atributos para Internet das Coisas

Antonio L. Maia Neto[1], Stephen E. Richardson[2], Mark Horowitz[2], Leonardo B. Oliveira[1]

[1] Universidade Federal de Minas Gereais (UFMG)
[2] Stanford University, CA

{lemosmaia,leob}@dcc.ufmg.br, steveri@stanford.edu, horowitz@ee.stanford.edu

Abstract. *The integration of resource-constrained devices into IoT domains depends on a cryptographic foundation that enables the establishment of the essential security mechanisms. Attribute-based Signatures (ABS) are an elegant option to implement an authentication and authorization scheme. This work discusses different approaches to speed up ABS considering as the target platform the resource-constrained devices on IoT domains.*

Resumo. *Antes de integrar dispositivos com restrições de recursos computacionais em domínios de IoT, deve-se garantir que possuam ferramental criptográfico suficiente para estabelecer mecanismos de segurança essenciais. Neste contexto, as Assinaturas baseadas em Atributos (ABS) mostram-se uma opção elegante na implementação de um esquema de autenticação e autorização. Neste trabalho discutimos diferentes abordagens para a aceleração de (ABS) considerando-se como plataforma alvo os dispositivos com restrições de recursos computacionais em domínios de IoT.*

1. Introdução

A Internet das Coisas (*Internet of Things* – IoT), ainda que distante da previsão feita por Mark Weiser [Weiser 1991] em 1991, talvez seja hoje o conceito que melhor nos remeta à visão de computação pervasiva, ubíqua. De fato, IoT pode ser encarada como a consolidação da interconectividade de domínios computacionais totalmente diferentes entre si como, por exemplo, computação móvel e em nuvem, sistemas ciber-físicos, redes veiculares e de sensores sem fio, além, é claro, dos sistemas distribuídos tradicionais.

A consolidação dessa diversidade de contextos acaba por possibilitar a integração e centralização de serviços, o que traz benefícios significativamente altos aos usuários finais e, assim, cada vez mais domínios passam a fazer parte desta grande rede de coisas. Entretanto, a interconectividade de domínios tão distintos deve ser feita tendo a segurança digital como um de seus pilares fundamentais, já que sistemas antes isolados acabam por ser expostos ao iminente risco de ataques cibernéticos [Abdul-Ghani et al. 2018].

Dentro deste cenário tão heterogêneo, há uma classe de dispositivos que merece especial atenção em relação a segurança digital. Trata-se daqueles que tem limitações de recursos computacionais – baixo poder de processamento e comunicação e pequena capacidade de memória e armazenamento. Antes de integrar estes dispositivos em um domínio de IoT, deve-se garantir que possuam ferramental criptográfico suficiente para estabelecer os mecanismos de segurança essenciais de identificação, autenticação e autorização. Portanto, eles devem ser capazes de executar primitivas criptográficas no nível adequado

de segurança e, assim, implementar os protocolos necessários, caso contrário, tornam-se alvos frágeis de potenciais atacantes. Neste contexto, criptossistemas convencionais baseados em infraestrutura de chaves públicas (*Public Key Infrastructure* – PKI) são, muitas vezes, inviáveis, pois a autenticação, armazenamento, comunicação e gerenciamento de certificados digitais causam sobrecarga computacional significativa e os dispositivos com restrições de recursos computacionais, muitas vezes, não suportam [Oliveira et al. 2008].

A Criptografia baseada em emparelhamento (*Pairing-based Cryptography* – PBC) [Sakai et al. 2000] tem sido aplicada como alternativa à PKI no contexto de dispositivos com restrições computacionais [Oliveira and Dahab 2006, Aranha et al. 2009]. A principal aplicação possibilitada por PBC é a Criptografia baseada em Identidade (*Identity-based Cryptography* – IBC) [Shamir 1984]. Em IBC, as chaves públicas são derivadas de informações publicamente conhecidas *a priori* pelas entidades de um domínio e relacionam-se univocamente a cada uma destas entidades, o que elimina a necessidade de certificados digitais, melhorando o desempenho dos protocolos criptográficos por suprimir a necessidade de verificação e diminuir o uso de espaço para armazenamento.

A Criptografia baseada em Atributos (*Attribute-based Cryptography* – ABC) [Goyal et al. 2006], por sua vez, estende a ideia por traz de IBC, mas, ao invés da identidade das entidades, considera seus atributos. Uma das aplicação de ABC é o esquema de Assinaturas baseadas em Atributos (*Attribute-based Signature* – ABS) com o qual é possível implementar criptograficamente um mecanismo de autenticação e autorização baseado em atributos [Neto et al. 2016], possibilitando o gerenciamento do controle de acesso autenticado a recursos em dispositivos com restrições computacionais de um contexto de IoT. Entretanto, os tempos de execução de ABS nestas plataformas ainda são um impeditivo para sua aplicação prática.

Objetivo. Neste trabalho, apresentamos e discutimos diferentes abordagens, tanto em software quanto em hardware, de aceleração do cálculo de geração e verificação de assinaturas baseadas em atributos em dispositivos com restrições computacionais em um contexto de IoT.

Organização. O restante deste artigo é estruturado da seguinte forma: trabalhos relacionados à implementação eficiente de mecanismos de segurança em dispositivos com restrições de recursos computacionais são discutidos na Seção 2. Na Seção 3 apresentamos uma visão geral do esquema de ABS. Na Seção 4 apresentamos resultatos experimentais de diferentes estratégias de software para implementação do ABS. Na Seção 5 discutimos estratégias de aceleração (*speedup*) em hardware para do ABS. Concluímos na Seção 6.

2. Trabalhos Relacionados

A primeira implementação prática de PBC em dispositivo com restrições de recursos computacionais, um microcontrolador de 8 bits, 78 MHz e 4 KB de RAM, foi apresentada no trabalho *TinyTate* [Oliveira et al. 2007], onde o cálculo de um emparelhamento foi feito em 30 s.

Em *NanoECC*, Szczechowiak *et al.*, considerando 80 bits de nível de segurança, apresentaram uma melhora significativa no cálculo de um emparelhamento, cerca de 12 s, em um microcontrolador similar, porém equipado com mais memória RAM (10 KB).

Já Gouvêa e López [Gouvêa and López 2009], aumentando o nível de segurança para 128 bits e baseados em implementação de código especificamente construído para a plataforma MSP430 de 16 bits e 8 MHz, reportaram um tempo de cerca de 15 s no cômputo de um emparelhamento. Mais tarde, intensificando o nível de otimização e lançando mão de multiplicador dedicado em hardware, Gouvêa *et al.* [Gouvêa et al. 2012] reportaram tempos de cerca de 10 s e 6 s, respectivamente.

Como pode ser vistos nos resultados alcançados em plataformas de 8 e 16 bits, os tempos para computação de emparelhamentos não são adequados para contextos de IoT, principalmente porque protocolos interativos, como em autenticação, requerem múltiplos cálculos emparelhamentos. Assim, trabalhos subsequentes consideram plataformas de 32 bits e apresentam uma melhora significativa nesses valores, por exemplo, 164 ms em 128 bits de nível de segurança [Unterluggauer and Wenger 2014] e 74,5 ms em 80 bits de nível de segurança [Neto et al. 2016]. Ainda assim, como em ABS há necessidade de envolver produto de emparelhamentos, é necessário buscar opções ainda mais rápidas.

3. Assinaturas baseadas em Atributos em IoT

Em um domínio onde recursos são compartilhados entre entidades que podem ser agrupadas em conjuntos baseados nos seus atributos comuns, as relações de confiança são, geralmente, estabelecidas por grupos em detrimento de relações individuais. É esse o caso de IoT, onde dispositivos que oferecem serviços tem como clientes grupos de entidades e aplicar um esquema de controle de acesso discricionário pode inviabilizar a manutenção e gerenciamento da autorização.

Um esquema de ABS pode ser diretamente aplicado neste contexto para implementar um esquema autenticado de autorização baseada em atributos [Neto et al. 2016]. Tal esquema pode ser descrito como se segue: *i*) as chaves privadas de uma entidade do domínio refletem seu conjunto de atributos; *ii*) as políticas de acessos a recursos, chamadas predicados, por sua vez, relacionam um (sub)conjunto de atributos do domínio em funções lógicas *booleanas*; *iii*) uma assinatura estabelecida com o (sub)conjunto de chaves (atributos) que satisfaçam um predicado pode ser verificada e, caso a verificação seja exitosa, o acesso pode ser concedido à entidade em questão.

Entretanto, apesar de um esquema de ABS teoricamente resolver o problema de controle de acesso em IoT, os números experimentais apresentados até então (cerca de 1,5 s e 3,0 s para geração e verificação de assinaturas, respectivamente), inviabilizam sua aplicação prática. Nas próximas seções discutimos estratégias, tanto em software quanto em hardware, de *speedup* do ABS.

4. *Speedup* do ABS em Software

Implementação. Para avaliar o *speedup* de ABS em software, reproduzimos a implementação apresentada no trabalho de Neto *et. al.* [Neto et al. 2016], construída sobre a biblioteca criptográfica RELIC[1], cuja construção de PBC envolve aritmética de curvas elípticas definidas sobre corpos primos[2].

[1] `https://github.com/relic-toolkit/relic`

[2] Curva BN-254 [Barreto and Naehrig 2005] de grau de mergulho 12, definida sobre o corpo primo de 254 bits, ordem $n(u) = 36u^4+36u^3+18u^2+6u+1$, característica $p(u) = 36u^4+36u^3+24u^2+6u+1$, parametrizadas pelo inteiro $u = -(2^{62}+2^{55}+1)$ e cerca de 100 bits de nível de segurança [Joux 2013].

Plataformas Experimentais. Como plataformas experimentais utilizamos dois dispositivos: a plataforma de IoT alvo, um Arduino Due, com microcontrolador baseado na arquitetura ARM Cortex-M3 de 32 bits e 84 MHz, com 96 KB de memória RAM e 512 KB de memória *flash*; e uma plataforma auxiliar, Raspberry Pi 3 B, baseada em um processador de quatro núcleos Cortex-A53 de 64 bits e 1.2 GHz, com 1 GB de memória RAM e armazenamento expansível. O objetivo da plataforma alvo é avaliamos experimentalmente os tempos de execução de diferentes versões de implementação dos algoritmos de ABS no contexto de IoT. Já a plataforma auxiliar, por ser baseada em arquitetura similar à plataforma alvo, porém com maior poder computacional, é usada para analisar o comportamento dos algoritmos através do *profiling* de execução.

Experimentos. O foco de nossa análise são os algoritmos de geração e verificação de assinaturas baseadas em atributos, já que são aqueles executados pelos dispositivos de IoT [3]. Lançando mão, com pequenas necessidades de adaptação do código, de duas versões da aritmética de corpos primos presentes na biblioteca criptográfica, construímos três variações de código[4]: linguagem `C` – aritmética de corpos primos totalmente implementada em `C`, *assembly* – algumas funções da aritmética de corpos primos implementadas em código *assembly* ARM, e *overclock* – versão *assembly* executada sobre o microcontrolador do Arduino Due em um estado de *overclock*[5].

Resultados Experimentais. A versão em linguagem `C` apresentou tempos de execução de 3,28 s para geração e 6,434 s para verificação de uma assinatura. Já na versão *assembly* os tempos foram reduzidos para, 1,59 s e 2,93 s, que representam *speedup* de 2,06 e 2,2, respectivamente.

Baseados nos primeiros resultados práticos e considerando-se a melhora no tempo de execução proporcional ao aumento da frequência de *clock* de 84 MHz para 114 MHz, estabelecemos os valores teóricos aproximados para geração e verificação de uma assinatura de, respectivamente, 1,17 s e 2,16 s. No entanto, os resultados experimentais foram, respectivamente, de 1,23 s e 2,268 s. Dado que, entre outros fatores, a melhora no tempo de execução não é proporcional ao aumento da frequência e que há dificuldades na medida de tempo de execução em um microcontrolador em *overclock*, consideramos razoável a diferença de aproximadamente 100 ms.

Os valores experimentais nos mostram, então, um *speedup* de aproximadamente 2.7 para geração e 2.8 para verificação de assinaturas em relação à versão em linguagem `C`, ainda assim, acreditamos que para viabilizar a aplicação de ABS na prática, os resultados deveriam ser inferiores a 1,0 s para ambos algoritmos, o que demandaria valores de *speedup* de cerca de 3,3 e 6,5, respectivamente.

Comportamento de ABS. A execução dos algoritmos de ABS na plataforma alvo, aliadas à análise de comportamento do mesmo via uma ferramenta de *profiling* de execução como `Gprof` [Graham et al. 1982], pode nos mostrar qual o nível de otimização alcançado pela implementação das funções em *assembly* ARM e, inclusive, entender onde devemos focar os esforços para acelerar o esquema de ABS. Como a plataforma alvo, por se basear em um microcontrolador com limitações, impede o uso de tal ferramental, utilizamos a

[3] O *setup* do criptossistema e a geração de chaves são executados, na prática, por equipamentos grande poder computacional.

[4] O predicado considerado nos experimentos é da forma $A \wedge B$, ou seja, uma operação *booleana* E ($\wedge$) que relaciona dois atributos.

[5] De 84 MHz para 114 MHz, que é a frequência máxima de operação estável do microcontrolador.

plataforma auxiliar.

A Tabela 1 apresenta os resultados do *profiling* de código para as funções mais custosas encontradas na execução dos algoritmos de ABS, são elas, a função de multiplicação de dois elementos do corpo primo (`fp_muln_low`) e a função de redução de um elemento ao módulo do número primo (`fp_rdcn_low`). Nas colunas "% Execução" temos a contribuição da função no tempo de execução total do algoritmo. Combinando o resultado desta coluna com os resultados experimentais das execuções na plataforma alvo, podemos estimar, na coluna "Tempo (s)", o tempo de execução de cada uma dessas funções.

Algoritmo ABS	Função	Linguagem C		*Assembly* ARM	
		% Execução	Tempo (s)	% Execução	Tempo (s)
Geração de Assinatura	`fp_muln_low`	33.65%	1.10	23.81%	0.38
	`fp_rdcn_low`	32.83%	1.08	20.48%	0.33
Verificação de Assinatura	`fp_muln_low`	33.95%	2.18	27.05%	0.79
	`fp_rdcn_low`	25.28%	1.63	20.48%	0.48

Tabela 1. *Profiling* da execução da geração e verificação de uma assinatura ABS.

Dos resultados apresentados, podemos calcular que a média de *speedup* alcançada pela implementação em *assembly* das funções mais custosas foram de 3,08 para a geração e 3,06 para a verificação de uma assinatura. Isto significa que, mesmo que apliquemos este nível de *speedup* para todo o restante do código, ainda assim o resultado ficaria abaixo da adoção prática do esquema de ABS. Por isso, na próxima seção discutimos algumas abordagens em hardware para buscar o nível de *speedup* adequado.

5. *Speedup* do ABS em Hardware

Adição de instruções ao conjunto de instruções da plataforma alvo. Imaginemos que fosse possível adicionar à nossa arquitetura alvo duas instruções para substituir as funções mais custosas de ABS. A Tabela 1 apresenta a estimativa de tempo de execução para tais funções (`fp_muln_low` e `fp_rdcn_low`) é de, respectivamente, 0,38 s e 0,33 s na geração e 0,79 s e 0,48 s na verificação de assinatura. Agora, baseado no argumento de Amdahl [Amdahl 1967], se estas duas instruções executassem em tempo desprezível, os tempos de execução seriam de 0,89 s e 1,54 s para geração e verificação de assinatura, respectivamente. Ou seja, o esforço de implementação de novas instruções para na plataforma alvo não seria suficiente para se alcançar tempos adequados para o ABS em IoT.

Desenvolvimento de hardware dedicado. As questões que envolvem desenvolvimento de hardware dedicado à execução de aplicações específicas vão desde o grau de *speedup* necessário e requisitos de eficiência energética a custos de produção e prototipação. Por exemplo, em circuitos integrados para uma aplicação específica (*Application Specific Integrated Circuit* – ASICs), o nível de *speedup* e eficiência energética seriam ideais para um contexto escalável como IoT. Entretanto, os custos envolvidos para produção e prototipação são proibitivos. A alternativa mais viável são as *Field Programmable Gate Array* (FPGAs), onde conseguimos alcançar um *speedup* adequado ao contexto de IoT com custos bem menores e maior flexibilidade para simulação e prototipação.

6. Conclusão e Trabalhos Futuros

Os resultados experimentais dos algoritmos de geração e verificação de assinaturas de ABS mostram que mesmo que o *speedup* alcançado via software, se aplicado a todo o

código não seria suficiente para viabilizar a execução de ABS em um contexto de IoT. Além disso, também é possível observar que mesmo a estratégia em hardware de adição de instruções especiais ao conjunto de instruções da arquitetura alvo não adequaria o ABS a um domínio de IoT. Portanto, uma estratégia de projeto de hardware dedicado, como FPGA, se faz necessária e será adotada nos trabalhos futuros.

Referências

Abdul-Ghani, H. A., Konstantas, D., and Mahyoub, M. (2018). A comprehensive IoT attacks survey based on a building-blocked reference model. *IJACSA'18*.

Amdahl, G. M. (1967). Validity of the single processor approach to achieving large scale computing capabilities. In *SJCC'67*.

Aranha, D. F., Oliveira, L. B., Lopez, J., and Dahab, R. (2009). NanoPBC: implementing cryptographic pairings on an 8-bit platform. In *CHiLE'09*.

Barreto, P. S. L. M. and Naehrig, M. (2005). Pairing-friendly Elliptic Curves of Prime Order. In *SAC'05*.

Gouvêa, C. P., Oliveira, L. B., and López, J. (2012). Efficient software implementation of public-key cryptography on sensor networks using the MSP430X microcontroller. *JCEN'12*.

Gouvêa, C. P. L. and López, J. (2009). Software implementation of pairing-based cryptography on sensor networks using the MSP430 microcontroller. In *Indocrypt'09*.

Goyal, V., Pandey, O., Sahai, A., and Waters, B. (2006). Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data. In *CCS'06*.

Graham, S. L., Kessler, P. B., and Mckusick, M. K. (1982). Gprof: A call graph execution profiler. In *SIGPLAN'82*.

Joux, A. (2013). A new index calculus algorithm with complexity $l(1/4 + o(1))$ in small characteristic. In *SAC'13*.

Neto, A. L. M., Souza, A. L. F., Cunha, I., Nogueira, M., Nunes, I. O., Cotta, L., Gentille, N., Loureiro, A. A. F., Aranha, D. F., Patil, H. K., and Oliveira, L. B. (2016). AoT: Authentication and Access Control for the Entire IoT Device Life-Cycle. In *Sensys'16*.

Oliveira, L. B., Aranha, D., Morais, E., Daguano, F., López, J., and Dahab, R. (2007). TinyTate: Computing the tate pairing in resource-constrained nodes. In *NCA'07*.

Oliveira, L. B. and Dahab, R. (2006). Pairing-based cryptography for sensor networks. In *NCA'06*.

Oliveira, L. B., Scott, M., Lopez, J., and Dahab, R. (2008). TinyPBC: Pairings for Authenticated Identity-Based Non-Interactive Key Distribution in Sensor Networks. In *INSS'08*.

Sakai, R., Ohgishi, K., and Kasahara, M. (2000). Cryptosystems Based on Pairing. In *SCIS'00*.

Shamir, A. (1984). Identity-based cryptosystems and signature schemes. In *CRYPTO'84*.

Unterluggauer, T. and Wenger, E. (2014). Efficient pairings and ECC for embedded systems. In *CHES'14*.

Weiser, M. (1991). The computer for the 21 st century. *Scientific American*.