Lista de Certificados Revogados Limitada

Lucas Ferraro, Ricardo Felipe Custódio

[1]Departamento de Informática e Estatística (INE)
Universidade Federal de Santa Catarina (UFSC)
Florianópolis – SC – Brasil

{lucas.ferraro,custodio}@inf.ufsc.br

Abstract. *Certificate Revocation Lists represent a common solution to the problem of digital certificate revocation. When this solution is applied on environments with thousands of certificates it can reach an unmanageable size. It can result in more than 90% of the digital signature's size. To offer a size control we propose the Limited Certificate Revocation List. With this mechanism is possible to plan the final size of the Certificate Revocation Lists.*

Resumo. *Listas de Certificados Revogados representam uma solução comum para o problema da revogação de certificados digitais. Quando essa solução é usada em ambientes com milhares de certificados seu tamanho pode tornar-se algo inviável. Resultando em mais de 90% do tamanho de uma assinatura digital. Para oferecer um controle desse tamanho foi elaborada a Lista de Certificados Revogados Limitada. Com ela é possível projetar o tamanho final da Lista de Certificados Revogados.*

1. Introdução

A evolução da ciência da computação está mudando o mundo, e até mesmo os processos mais tradicionais estão sendo repensados de acordo com a evolução tecnológica. Isso inclue a migração de documentos reais (documentos em geral, cartão de identidade, contratos, etc) para o formato digital.

É necessário afirmar a autoria e a integridade desses documentos eletrônicos. Para isso assinaturas digitais foram desenvolvidas com os mesmos propósitos da assinatura manuscrita. Mas outras questões surgem com as assinaturas digitais. E uma dessas questões é como confirmar a identidade de um assinante. Essa confiança é oferecida pela Infraestrutura de Chaves Públicas (ICP).

Em uma ICP, cada entidade é identificada por certificados digitais. Certificado Digital é uma estrutura de dados que une a identificação de uma entidade (por exemplo o número do CPF) com sua chave pública. Todavia, um certificado digital pode perder sua validade e precisar ser revogado. O processo de revogação é conhecido como um dos maiores desafios da ICP. A forma mais comum de revogação é pela Lista de Certificados Revogados (LCR).

Para oferecer um controle do tamanho da LCR foi elaborada a Lista de Certificados Revogados Limitada (LCRL). A LCRL usa a mesma estrutura da LCR, mas aplica alguns procedimentos para emitir certificados. Esses procedimentos limitam a quantidade de certificados que apontam para a mesma LCR. Assim é possível projetar o tamanho máximo que uma LCR pode chegar.

Esse trabalho tem a seguinte organização: Seção 2 traz uma visão geral dos mecanismos de revogação; Seção 3 apresenta os detalhes da Lista de Certificados Revogados Limitada; Seção 4 apresenta uma análise comparativa da LCR e da LCRL e uma análise do comportamento de uma assinatura com a LCRL; Na Seção 5 são apresentadas as considerações finais.

2. Trabalhos Relacionados

Mecanismos de revogação são uma característica fundamental das ICPs. Pois, um certificado tem uma data de validade. E a maioria dos certificados duram até essa data. Contudo um certificado pode perder sua validade antes da data de expiração. E, quando essa validade é comprometida deve-se ter uma maneira de notificar as outras entidades da ICP sobre esse fato. Além dessa notificação, é necessário que a Autoridade Certificadora (AC) publique tal informação. A publicação dessa informação é conhecida como mecanismo de revogação. Existem dois mecanismos principais: aqueles que usam estruturas de dados (listas, árvores, etc); e aqueles que defendem que a revogação não deveria existir.

2.1. Mecanismos Baseados em Listas

A Lista de Certificados Revogados é considerada a forma mais simples de revogação. A LCR, definida por Cooper et al. [Cooper et al. 2008] é um padrão bem estabelecido e disseminado. Em *International Telecommunication Union* (ITU) [ITU 2008] é apresentada a delta-LCR que tem como objetivos: a redução do tamanho da LCR. Feito pela publicação das diferenças entre a nova LCR e última, e; a redução do período de publicação de LCRs. Feito pela publicação de delta-LCRs entre as LCRs base. Também apresentada por ITU [ITU 2008], a LCR *Distribution Points* oferece a possibilidade de dividir a LCR em grupos de certificados revogados com a mesma razão de revogação.

Cooper [Cooper 1999] criou dois mecanismos baseados em listas, as LCRs Sobrepostas, que visam reduzir a largura de banda necessária a cada atualização mantendo mais de uma LCR válida ao mesmo tempo, e as LCRs Segmentadas, que visam reduzir o tamanho da LCR dividindo-a em segmentos. A desvantagem dessa abordagem é saber em qual segmento o certificado aparecerá. Isso ocorre pelo fato do algoritmo, que insere o certificado na lista, ser aleatório. Cooper [Cooper 2000] tentou aprimorar as delta-LCRs adicionando a ideia das LCRs Sobrepostas. Esse novo mecanismo foi chamado de *Sliding Window Delta-CRL*. Entretando esses mecanismos implicam no aumento da complexidade no processo de verificação de uma assinatura digital.

Goyal [Goyal 2007] propôs particionar a LCR em conjuntos de certificados com número de série consecutivos. Para revogar uma partição inteira é necessário incluir apenas o primeiro número de série da partição. Esse modelo altera o algoritmo de verificação de certificados e faz com que o verificador conheça as informações das partições.

McDaniel e Jamin [McDaniel and Jamin 2000] deram ao certificado um período após sua emissão em que o certificado não pode ser revogado. Além disso criaram uma janela de revogação, tempo o qual um certificado pode ser revogado. O benefício desse método é a redução da LCR, pois tem menos certificados e por menos tempo. Segundo McDaniel e Jamin, o verificador pode não conseguir acessar a LCR adequada para verificar o certificado e ser forçado a considerar o certificado não confiável.

2.2. Mecanismos Baseados em Funções Hash e Árvores

A *Certification Revocation Trees* (CRT), proposta por Kocher [Kocher 1998], tem como objetivo diminuir o tamanho do artefato de revogação. Mas, apesar de ter um tamanho mínimo, o processamento necessário para gerar a árvore a cada atualização e o caminho de validação a cada requisição é muito custoso para a AC. Solworth [Solworth 2008] resolveu esse problema forçando os usuários a processarem a árvore, mas fez o verificador ser obrigado a ter uma cache com todos os certificados revogados e manter-se conectado à internet em tempo integral.

2.3. Mecanismos Sem Artefatos de Revogação

Rivest [Rivest 1998] e Scheibelhofer [Scheibelhofer 2005] propuseram certificados de curta duração para evitar os artefatos de revogação, mas sobrecarregam a AC com pedidos de emissão. Já Vigil e Custódio [Vigil and Custódio 2012] propuseram a reemissão de certificados para alterar a data final de validade para um momento antes do pedido de revogação. E como consequência exige a alteração do padrão dos certificados, pois na reemissão o certificado altera o hash.

2.4. Outros Trabalhos Relacionados

Trabalhos como Ofigsbo et al. [Ofigsbo et al. 2010] dizem que não há um mecanismo de revogação ideal para todos os cenários. Esses trabalhos apresentam *frameworks* e modelos matemáticos para avaliação de quão seguro e custoso cada mecanismo é para cada cenário específico.

3. Lista de Certificados Revogados Limitada - LCRL

A LCRL tem a mesma estrutura da LCR, mas são aplicadas algumas características de gerência. A primeira modificação é quanto ao processo de acumulação de certificados na LCR. Nenhum certificado será removido da LCR, mesmo depois da data de expiração do certificado. Com isso sempre será possível acessar a informação de revogação de um certificado com a última LCR emitida. A segunda modificação é referente à quantidade de LCRs, pois uma AC poderá ter mais de uma LCR. Os certificados que apontam para a mesma LCR serão limitados de forma a forçar a AC a ter mais de uma LCR. Ao alcançar o limite de certificados por LCR a AC deverá executar um processo de renovação. Dependendo da política da AC, o processo pode, além de renovar a LCR, renovar o par de chaves da AC.

A validação de longo prazo não precisa mais da LCR na estrutura da assinatura. Se o verificador puder acessar a última LCR emitida pela AC, então ele pode verificar se o certificado era válido quando a assinatura foi feita. A única restrição deve-se ao fato de que a LCR deve ter a data *thisUpdate* posterior à data da assinatura. Com isso podemos ver que, enquanto a LCR estiver disponível e o verificador puder baixá-la, não é necessário incluir a LCR na assinatura.

3.1. Renovação da AC

Exitem três métodos de renovação da AC.

Renovação por *Distribution Point*: A AC altera o *Distribution Point* (DP) (ponteiro para uma LCR específica) para uma nova LCR. Assim os próximos certificados a serem emitidos apontarão para a nova LCR. Esse é um processo simples, mas requer atenção ao controle do número de LCRs. Nesse método, um par de chaves será responsável por emitir uma ou mais LCRs. A quantidade de certificados relativos a cada LCR deve ser a mesma. Essa quantidade deve ser especificada pela política da AC;

Renovação por Par de Chaves: A AC troca o par de chaves e também altera seu DP. Esse processo é complexo, pois requer que a AC superior emita mais certificados para a mesma AC. Cada certificado é versionado e todos são funcionais. Apenas um certificado é usado pela AC para emitir certificados. A quantidade de certificados relativos a cada par de chaves deve ser a mesma. Essa quantidade deve ser especificada pela política da AC;

Renovação Completa: A AC tem mais de uma LCR para cada par de chaves. Primeiro, deve-se renovar o DP quantas vezes for especificado pela política da AC (esse valor especifica a quantidade de LCRs que um par de chaves pode gerenciar). Assim que atingir o valor máximo de renovações, é necessário renovar o par de chaves. Conforme o método Renovação por Par de Chaves. Para a renovação do DP são adotados os mesmos procedimentos do método Renovação por *Distribution Point*.

3.2. Repositório da AC

Um novo componente deve ser adicionado para ajudar no serviço de controle do tamanho da LCR. Esse componente será chamado de repositório da AC. Ele é um conjunto de fluxos de controle, estrutura de dados e métodos.

Estruturas de Dados:

$Cert_{index}$**:** indica o número de certificados emitidos para a LCR atual. A LCR atual é aquela que ainda aceita certificados para serem relacionados;

CRL_{index}**:** é o índice da LCR atual;

Key_{index}**:** é o índice do par de chaves atual. O par de chaves atual é aquele em uso para emitir novos certificados;

Key_{list}**:** é a lista de pares de chaves disponíveis para a AC. Todo par de chaves tem um certificado correspondente;

CRL_{list}**:** é a lista de todas as LCRs da AC. Toda LCR deve ser atualizada enquanto a AC continuar funcionando. O processo de atualização da LCR termina quando todos os seus certificados relacionados expiram, até mesmo os revogados;

$Cert_{limit}$**:** é o número máximo de certificados que apontam para a mesma LCR como DP;

CRL_{limit}**:** é o número máximo de LCRs para cada par de chaves;

$IssueData$**:** é a composição do par de chaves atual e do DP da LCR atual. Esse objeto precisa ser atualizado quando a AC renova seu par de chaves ou sua LCR.

Métodos:

$checkCertLimit()$**:** verifica se a LCR atual chegou ao número máximo de certificados que podem ser relacionados a ela. A condição é: $Cert_{index} \leq Cert_{limit}$;

$getIssueData()$**:** retorna o $IssueData$ da AC. O $IssueData$ é usado para emissão de certificados;

$acHasCrlLimit()$**:** verifica se a AC tem ou não algum limite para o número máximo de LCRs. Por padrão esse método retorna falso;

$checkCrlLimit()$**:** verifica se o par de chaves atual chegou ao número máximo de LCRs. O número máximo de LCRs é definido pela política da AC;

$renewKeyPair()$**:** renova o par de chaves da AC. Nesse método a AC troca seu par de chaves e também seu certificado. Por consequência a AC também renova sua LCR, via $renewCrl()$ e zera o CRL_{index} via $renewCertIndex()$. O novo par de chaves deve estar no repositório antes dessa operação. O Key_{index} deve ser incrementado.

$renewCrl()$**:** renova a LCR. Esse método envolve, além da mudança da LCR atual, a zeragem do $Cert_{index}$. Cada LCR nova é registrada no repositório da AC e o CRL_{index} é incrementado pelo método $updateCertIndex()$;

$register(Cert)$**:** registra o certificado no repositório da AC. Quando esse método é executado, o repositório deve ser atualizado pelo método $updateRepository()$;

$updateCertIndex()$**:** atualiza o $Cert_{index}$ via $Cert_{index} + 1$;

$renewCertIndex()$**:** define zero no $Cert_{index}$ via $Cert_{index} = 0$;

$updateCrlIndex()$**:** atualiza o índice da LCR. O novo valor definido é o índice que aponta para a próxima LCR a ser usada. Esse índice deve seguir as regras da estrutura de dados utilizada para as LCRs, como por exemplo: $LCR = \{LCR_1, LCR_2, \ldots, LCR_{n-1}, LCr_n, LCR_{n+1}, \ldots, LCR_z\}$, onde $n = CRL_{index}$ (representa o índice da LCR atual). Quando pegar uma nova LCR é necessário incrementar o índice via $n = n + 1$. A LCR_z é a última LCR permitida para o par de chaves atual ($z = CRL_{limit}$);

$renewCrlIndex()$**:** zera o CRL_{index} via $CRL_{index} = 0$;

$updateRepository()$**:** atualiza os índices e verifica se a renovação da LCR, usando $renewCrl()$, e a renovação do par de chaves, usando $renewKeyPair()$, são necessárias.

Fluxo de Controle da Emissão de Certificados: Para alcançar o objetivo de limitar o tamanho da LCR, além de métodos e variáveis de controle, é necessário oferecer um fluxo de controle. Esse fluxo pode ser visualizado na Figura 1.

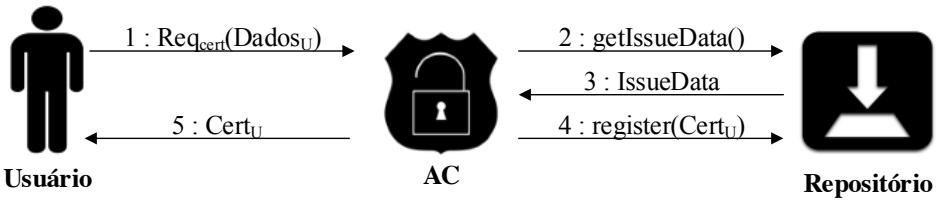

Figura 1. Fluxo de controle da emissão de certificados.

Passo 1: O usuário (entidade final) envia uma requisição de certificado para a AC. Tal requisição inclui os dados do usuário, tal como, chave pública e informações de identificação;

Passo 2: A AC pede ao repositório seus dados de emissão para emitir o certificado requisitado;

Passo 3: O repositório responde com os dados de emissão da AC;
Passo 4: A AC gera o certificado do usuário e registra no repositório. Esse passo aciona a execução do método $updateRepository()$. Como consequência o repositório ficará pronto para responder a uma nova requisição;
Passo 5: A AC envia o certificado para o usuário;

4. Análise

A análise apresentada será baseada nas diferenças entre a LCR e a LCRL nos quesitos: tamanho, validação de uma assinatura digital, adição na estrutura da assinatura e suporte da AC.

Tamanho da LCR: Pela fórmula, $LCR = 65 + \sum_1^n(x_n + 10) + (19 + s) * r + E + \frac{chave}{8}$, pode-se calcular o tamanho da LCR em bytes. Onde: $\sum_1^n(x_n + 10)$ significa o somatório dos nomes da estrutura do nome do emissor (*Country*, *CommomName*, etc), onde x_n é a menor potência de dois possível para representar o nome e 10 representa a estrutura para montar o nome; $(19 + s)$ indica cada certificado revogado, pois 19 é a soma do tempo e da estrutura para suportar o número de série do certificado, e s é o tamanho do número de série em bytes (todos os certificados de uma AC têm o mesmo tamanho de número de série); r é a quantidade de certificados revogados; E representa as estensões e campos opcionais da LCR; chave é o comprimento da chave em bits, assim deve-se transformar em bytes. Para efeito comparativo entre a LCR e a LCRL vamos adotar os seguintes valores: $x_1 = 5$ bytes, $x_2 = 4$ bytes , $s = 4$ bytes, $E = 0$ bytes e $chave = 2048$ bits.

Para efetuar a comparação será usada uma AC com 5 anos de duração e seus certificados emitidos terão 2 anos de validade. A taxa de revogação de certificados durante sua vida será de 10% (baseada em Hormann et al. [Hormann et al. 2006]). Essa AC irá emitir 200.000 certificados por ano.

A Tabela 1 apresenta o comportamento da LCR no cenário apresentado. Nota-se que a LCR não mantém os certificados expirados em sua estrutura. E por essa razão seu tamanho estabiliza-se a partir do segundo ano [Ma et al. 2006]. Devido a essa estratégia, ainda, é possível observar que a LCR contém apenas $7,5\%$ de todos os certificados válidos, ou seja, menor porcentagem que a taxa de revogação.

Tabela 1. Demonstração de crescimento da LCR.

Ano	Total[1]	Revogados[1]	Expirados[1]	Não Expirados[1]	CRL[1]	%	Tam.[2]
1	200	10	0	200	10	25	230.350
2	400	20	0	400	30	7,5	690.350
3	600	20	200	400	30	7,5	690.350
4	800	20	200	400	30	7,5	690.350
5	1000	20	200	400	30	7,5	690.350

[1] Os valores apresentados em milhares.
[2] Tamanhos em bytes.

Para a LCRL será necessário adotar mais dois valores ao cenário, o número de certificados relacionados a uma LCRL, que será 50.000 e o número de LCRLs por par de chave da AC, que será 10. Será adotado o método de Renovação Completa da AC. De

acordo com esse cenário, tem-se 50.000 multiplicado pela taxa de revogação 10%, o que representa 5.000 certificados revogados por LCRL ou 115.350 bytes. Pela Tabela 1 a LCR atinge 690.350 bytes, com isso pode-se afirmar que o uso da LCRL gera uma redução de $83,29\%$ no tamanho da LCR, mas esses valores podem variar dependendo dos valores adotados para a LCRL.

Durante a operação da AC com a LCRL outros valores podem ser observados, pois a cada 50.000 certificados emitidos uma renovação é exigida. Ao completar cinco anos de operação a AC terá vinte LCRLs (quatro novas por ano), dois pares de chave (o segundo será utilizado a partir do terceiro ano). No quinto ano de atividade o primeiro par de chaves não precisará mais ser usado para atualizar LCRLs, pois todos os certificados emitidos por ela estarão revogados.

Validação de uma assinatura: A validação de uma assinatura digital com LCRL é igual à LCR. A diferença é que sempre é possível usar a LCRL atual para verificar a assinatura, pois ela tem toda a informação necessária. Para prova temporal ainda é necessário a adição de carimbos do tempo.

Adição na estrutura da assinatura: A assinatura não precisa conter a LCRL na sua estrutura enquanto os algoritmos criptográficos, que a assinatura usa, forem seguros. Com isso, é possível manter as assinaturas com um tamanho mínimo. Mesmo quando necessário adicionar as informações de revogação, o impacto da LCRL será bem menor que o impacto da adição de uma LCR.

Suporte da AC: A AC deverá incluir em seu software um conjunto de funções e estruturas de dados para suportar a LCRL. O controle de seu tamanho demanda um gerenciamento de um conjunto de LCRLs e um conjunto de pares de chaves. A AC deve oferecer atualizações contínuas das listas, até mesmo quando nenhum certificado novo é relacionado à ela. Mas a AC para de atualizar uma determinada LCRL quando todos os seus certificados relacionados expiram.

5. Conclusão

Para oferecer à AC a opção de limitar o tamanho da LCR, foram modificados alguns conceitos do modelo atual. O principal conceito alterado foi quanto a definição da AC, na sua forma de representação. Pois no modelo proposto a AC pode ter sua identidade relacionada a um conjunto de pares de chaves. Adicionalmente foi inserida uma estrutura para habilitá-la a gerenciar essas chaves.

Essas modificações trazem dois benefícios para as assinaturas digitais. O primeiro, é relacionado ao tamanho das assinaturas ao adicionar a LCR. O segundo, é a forma de uso da LCR no processo de validação de uma assinatura. Pois, equanto os algoritmos usados na assinatura e na LCR forem válidos será possível utilizar a última LCR emitida para validar a assinatura.

Foi tomado como objetivo o controle do tamanho da LCR nesse trabalho. Assim, a largura de banda para os momentos de atualização da LCR e a periodicidade de atualização da LCR são problemas conhecidos, mas não tratados nesse trabalho.

Como trabalho futuro poderia ser elaborada uma análise estatística sobre o tamanho máximo aceitável para um LCR comparada com o número máximo de LCRs e pares de chaves que uma AC pode gerenciar.

Referências

Cooper, D. (1999). A model of certificate revocation. In *Computer Security Applications Conference (ACSAC '99) Proceedings. 15th Annual*, pages 256 –264, Phoenix, USA.

Cooper, D. (2000). A more efficient use of delta-crls. In *Proceedings of the 2000 IEEE Symposium on Security and Privacy (SP'00)*, pages 190 –202, Washington, DC, USA.

Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and Polk, W. (2008). Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280 (Proposed Standard).

Goyal, V. (2007). Certificate revocation using fine grained certificate space partitioning. In *Financial Cryptography and Data Security (FC'07)*, pages 247–259. Springer Berlin Heidelberg, Scarborough, Trinidad and Tobago.

Hormann, T. P., Wrona, K., and Holtmanns, S. (2006). Evaluation of certificate validation mechanisms. *Computer Communications*, 29(3):291–305.

ITU, I. T. U. (2008). Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks. Series x: Data networks, open system communications and security directory, International Telecommunication Union, Switzerland, Geneva. ITU-T Recommendation X.509.

Kocher, P. (1998). On certificate revocation and validation. In Hirchfeld, R., editor, *Financial Cryptography*, volume 1465 of *Lecture Notes in Computer Science*, pages 172–177. Springer Berlin Heidelberg.

Ma, C., Hu, N., and Li, Y. (2006). On the release of crls in public key infrastructure. In *Proceedings of the 15th conference on USENIX Security Symposium (USENIX-SS'06)*, volume 15, pages 17–28, Vancouver, Canada. USENIX Association.

McDaniel, P. and Jamin, S. (2000). Windowed certificate revocation. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, pages 1406 –1414 vol.3, Tel Aviv, Israel.

Ofigsbo, M., Mjolsnes, S., Heegaard, P., and Nilsen, L. (2010). Reducing the cost of certificate revocation: A case study. In Martinelli, F. and Preneel, B., editors, *Public Key Infrastructures, Services and Applications*, volume 6391 of *Lecture Notes in Computer Science*, pages 51–66. Springer Berlin Heidelberg.

Rivest, R. (1998). Can we eliminate certificate revocation lists? In Hirchfeld, R., editor, *Financial Cryptography*, volume 1465 of *Lecture Notes in Computer Science*, pages 178–183. Springer Berlin Heidelberg.

Scheibelhofer, K. (2005). Pki without revocation checking. In *4th Annual PKI R&D Workshop*, pages 48 – 61, NIST Gaithersburg MD, USA.

Solworth, J. (2008). Instant revocation. In Mjolsnes, S., Mauw, S., and Katsikas, S., editors, *Public Key Infrastructure*, volume 5057 of *Lecture Notes in Computer Science*, pages 31–48. Springer Berlin Heidelberg.

Vigil, M. A. G. and Custódio, R. F. (2012). Cleaning up the pki for long-term signatures. In *Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg 2012)*, pages 140–153, Curitiba, Brazil.