Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Um problema de escala na distribuição de chaves públicas e seu impacto na ICP Brasil

Ivan Menezes Sena

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientador
Prof. Dr. Pedro A. D. Rezende

Brasília
2017

Universidade de Brasília — UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação

Coordenador: Prof. Dr. Rodrigo Bonifácio de Almeida

Banca examinadora composta por:

Prof. Dr. Pedro A. D. Rezende (Orientador) — CIC/UnB
Prof. Dr. Jan Mendonca Correa — CIC/UnB
Prof. Dr. Edison Ishikawa — CIC/UnB

CIP — Catalogação Internacional na Publicação

Sena, Ivan Menezes.

Um problema de escala na distribuição de chaves públicas e seu impacto na ICP Brasil / Ivan Menezes Sena. Brasília : UnB, 2017.
219 p. : il. ; 29,5 cm.

Monografia (Graduação) — Universidade de Brasília, Brasília, 2017.

1. Criptografia assimétrica, 2. algoritmo RSA, 3. chave privada, 4. chave pública, 5. primos grandes, 6. ICP-Brasil, 7. MP 2200-2, 8. entropia, 9. geração de primos aleatórios, 10. ITI, 11. teste de Lenstra, 12. biblioteca GMP, 13. fastGCD

CDU 004

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília–DF — Brasil

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Um problema de escala na distribuição de chaves públicas e seu impacto na ICP Brasil

Ivan Menezes Sena

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Prof. Dr. Pedro A. D. Rezende (Orientador)
CIC/UnB

Prof. Dr. Jan Mendonca Correa
CIC/UnB

Prof. Dr. Edison Ishikawa
CIC/UnB

Prof. Dr. Rodrigo Bonifácio de Almeida
Coordenador do Bacharelado em Ciência da Computação

Brasília, 29 de novembro de 2017

Dedicatória

Dedico esta conquista aos meus amados pais Frederico Araújo Sena e Maria de Fátima Menezes Sena, minha estimada irmã Pâmela Menezes Sena Ferreira e meus dois queridos sobrinhos Eduardo Sena L. Ferreira e Maria Vitória Sena Ferreira.

Agradecimentos

À Universidade de Brasília, por me proporcionar a oportunidade de estudar e aprender com mestres, doutores e graduados das mais diversas áreas das ciências. Sou grato à cada membro do corpo docente, à direção e a administração dessa instituição de ensino.

À professora Germana Menezes da Nobrega, por todo auxílio e esforço empregados no meu regresso para a Universidade de Brasília.

Ao professor Pedro Antônio Dourado de Rezende, além de toda paciência e apoio durante a produção deste trabalho, suas aulas em Segurança de Dados me permitiram encontrar o meu lugar na Ciência da Computação.

Ao meu psicólogo, Thiago Cardoso Costa, por me auxiliar durante o meu período afastado da universidade e por me proporcionar as ferramentas didáticas necessárias para o meu sucesso acadêmico e profissional.

Ao meu colega Gabriel Gomes Gaspar por toda ajuda prestada durante as etapas iniciais deste trabalho.

Aos meus amigos Lucas Alem Martins e Fernanda Moraes, pela visão e revisão jurídica que serviram de apoio ao desenvolvimento deste trabalho.

À minha mãe Maria de Fátima Menezes Sena e à minha irmã Pâmela Menezes Sena Ferreira, por todo auxílio e tempo empreendidos na revisão ortográfica deste trabalho.

À minha família, por sempre me apoiar e sempre estar presente.

Aos meus amigos de infância, por nunca me deixaram duvidar das minhas capacidades.

Aos meus companheiros de farda, os Tubarões do Cerrado, por me motivarem a ser melhor todos os dias.

Resumo

O presente Trabalho de Graduação aborda um problema que surge na relação entre a função semiológica de irrefutabilidade de assinaturas digitais, e o desequilíbrio de riscos e responsabilidades legais imposto a signatários pelo regime da ICP-BR. Analisamos a relação lógica entre as premissas de inviolabilidade dessas assinaturas, e um nível adequado de entropia no processo de geração de números primos que compõem chaves criptográficas, sob condições de uso em larga escala do algoritmo RSA. Mostramos a implementação de um algoritmo estado-da-arte que relativiza tais premissas, pelo método indireto que permite derivar uma chave privada RSA a partir da chave pública correspondente, mediante fatoração do módulo desse par de chaves via cálculos amostrais de MDCs. Revelamos os resultados da execução desse algoritmo em um ambiente preparado para realizar teste de robustez em importantes acervos de chaves públicas, com vistas ao acervo das chaves certificadas no regime da ICP-BR. Desenvolvemos e apresentamos uma aplicação para teste de robustez de chaves públicas RSA individuais, voltada para o usuário comum. Expomos todas as tentativas frustradas de contactar ou engajar o ITI (Instituto Nacional de Tecnologia da Informação) e, à luz do recente caso "paralelo" na Estônia – em que 750 mil certificados digitais foram revogados devido à detecção de correspondente vulnerabilidade –, também motivos que supomos plausíveis para o desinteresse das autoridades responsáveis pela ICP-BR na cooperação originalmente proposta para este Trabalho.

Palavras-chave: Criptografia assimétrica, algoritmo RSA, chave privada, chave pública, primos grandes, ICP-Brasil, MP 2200-2, entropia, geração de primos aleatórios, ITI, teste de Lenstra, biblioteca GMP, fastGCD

Abstract

The present work examines a problem that arises from the relationship between the semiological irrefutability function of digital signatures and unbalances among risks and responsibilities imposed upon signatories under Brazil's official PKI regime (ICP-BR). We analyze the logical relation between digital signature's premises for inviolability and an adequate level of entropy for the process of generating pseudorandom prime integers to compose cryptographic keys, under conditions of widespread use of the RSA algorithm. We show an implementation of a state-of-the-art algorithm which relativizes these premises, through the indirect method that allows for the derivation of a RSA private key from the corresponding public key, upon factoring of the key pair's module through the sampling calculations of GCDs. We reveal the results from executing said algorithm on a setup prepared for testing robustness of important public key collections, aiming for the collection of RSA public keys certified under the ICP-BR regime. We expose all frustrated attempts to engage the National Institute of Information Technology (ITI) and, given the recent "parallel" case in Estonia – where 750 thousand digital certificates were revoked due to detection of the corresponding vulnerability –, also the motives we deem plausible for the unconcern and aloofness of ICP-BR authorities regarding the cooperation originally proposed for this work.

Keywords: Asymmetric cryptography, RSA algorithm, private key, public Key, big prime numbers, ICP-Brasil, MP 2200-2, entropy, generic prime number generation, ITI, Lenstra test, GMP library, fastGCD

Sumário

Lista de Figuras

Capítulo 1

Introdução

Em segurança nas comunicações, é possível observar que a busca por métodos para se estabelecer transmissões consideradas seguras entre dois agentes é um problema circular, do tipo metaforicamente conhecido em linguagem popular como "de ovo-e-galinha". Tal metáfora, assim empregada para ilustrar esta busca, se refere ao uso da criptografia para proteger sigilo ou integridade de mensagens ou documentos durante transmissões digitais (seja através do tempo, ou do espaço). Pois tal uso requer transmissão já protegida (no mínimo, com integridade) da chave criptográfica que antes encripta ou que depois autentica tais mensagens ou documentos. Este trabalho tematiza um problema relacionado à gerência de chaves criptográficas, que expõe nuanças dessa metáfora. Mais precisamente, um problema que surge com o uso disseminado de protocolos projetados para viabilizar a distribuição e gestão de chaves públicas em larga escala, operando sob o escopo de uma constelação de regras de natureza jurídica.

1.1 Apresentação de termos e conceitos

Antes de prosseguir com a introdução ao tema, é necessário uma breve apresentação de determinados termos, aos quais atribuiremos sentido técnico específico, adequado à correta interpretação do conteúdo deste trabalho. O primeiro termo que utilizaremos com uma acepção técnica específica, de extrema relevância para nossa abordagem, é "confiança". Confiança deve ser aqui entendida como aquilo que é essencial para um canal de comunicação e que não pode ser transferido da fonte para o destino através deste canal[1]. Aquilo que é essencial para que a informação supostamente transmitida faça sentido, ou seja, produza algum significado coerente com a situação cognitiva do

receptor da transmissão.

O segundo termo que usaremos com acepção específica é "premissas de confiança" [5]. Premissas de confiança são as condições ou requisitos para operação adequada dos mecanismos escolhidos para a proteção desejada. Exemplos incluem condições e requisitos relacionados à implementação, instalação e operação de determinado algoritmo criptográfico, de senha ou de chaves, garantia de origem de material criptográfico, canal utilizável para transmissões de inicialização destes, etc. Com tais definições, podemos abordar com mais clareza os processos de distribuição e gestão de chaves criptográficas, com foco em chaves públicas.

O terceiro termo a definir, é "assinatura digital". Assinatura digital é um esquema criptográfico criado para tornar possível a validação de origem e de integridade de mensagens e documentos em meio digital, de forma objetiva [2]; Ou seja, de forma que possa ser tecnicamente sustentada, no domínio jurídico, como oponível a terceiros. Em essência, a assinatura digital busca prover as mesmas funções semiológicas[2] que a assinatura de punho desempenha como meio de prova. A saber: inforjabilidade, inviolabilidade, irrecuperabilidade e irrefutabilidade[3]. A inforjabilidade remete à confiança do verificador na identificação de autoria, ou seja, na identificação correta do autor da assinatura. A confiança do verificador na integridade do conteúdo expresso no documento, vinculado à manifestação da vontade do signatário se for o caso, corresponde à inviolabilidade. A irrecuperabilidade é a convicção de ambos, verificador e signatário, de que a assinatura lavrada sobre um documento não pode ser reutilizada em outro documento, sem permitir detecção da manobra. Por fim, a irrefutabilidade, conceito técnico que, no domínio do Direito, ganha o nome de irretratabilidade: refere-se à confiança do verificador na inviabilidade técnica de negação da autoria da assinatura pelo

---

[1] Confiança não pode ser forçada (Gerk). Assim, quando um agente sente-se obrigado a agir como se confiasse noutro, tal situação indica potencial conflito de interesses, modelável pelas distintas percepções da extensão da confiança que ambos presumem do contexto, isto é, pelas diferenças entre a "confiança" que um agente presume ser demandada de si pelo outro, e a oferecida de si para o outro, relativamente ao assunto em tela e em expectativa a esse agir (ou ao não-agir). A definição de Confiança aqui adotada foi originalmente proposta por Edward Gerk no artigo "Toward Real-World models of Trust", disponível em http://mcwg.org/mcg-mirror/trustdef.htm

signatário[4].

A eficácia do esquema de assinatura digital decorre, e portanto depende, de três premissas de confiança:

1. Somente o titular de um par de chaves criptográficas assimétricas – pública e privada – deve controlar o uso de sua chave privada.
2. A titularidade de uma chave pública, usada para verificar assinaturas digitais do titular, ou para cifrar transmissões sigilosas destinadas a este, deve ser confiavelmente conhecida pelo verificador ou remetente.
3. O titular manifesta, ou pode manifestar, sua vontade no conteúdo de documentos assinados com sua chave privada.

A terceira premissa requer não só que a interpretação dos formatos digitais de documentos assinados seja invariante entre as interfaces computacionais utilizadas na lavra da assinatura e na verificação da mesma, mas também que os respectivos ambientes computacionais estejam sadios, isto é, livres de contaminação por programas maliciosos capazes de violar a característica "*wysiwyg*"[5]dessas interfaces, durante tais operações. Já a segunda premissa – confiança na titularidade de uma chave pública – tem sua demanda atendida, quando esta precisa superar problemas de escala, por protocolos de certificação digital, onde as premissas de sigilo e de integridade (de chaves privadas e de chaves públicas, respectivamente) são transformadas em autenticação recursiva com

---

[2] Na teoria linguistica cognitiva, conforme desenvolvida por Ronald Langacker por exemplo, a linguagem desempenha uma "função semiológica, que permite conceituações a serem simbolizadas por ilocuções e gestos", associada a uma "função interativa, que envolve comunicação, manipulação, expressividade e comunhão social" ([29], pg 14). No caso específico aqui citado, estamos nos referindo à conceituação de prova de autoria e/ou de manifestação de vontade, conforme a acepção doutrinária no Direito, expressa em documento eletronico que interage com atores através de passos – expressos por ilocuções e gestos – de um esquema de autenticação criptográfica (assinatura digital como procedimento), onde a autoria e/ou a manifestação da vontade de um signatário é simbolizada por componente descrito como autenticador criptográfico (assinatura digital como objeto).

[3] Aos níveis qualitativos considerados equiparáveis ou superiores aos do método da assinatura de punho como meio de prova de autoria, ou de eficácia probante para manifestação da vontade, na jurisprudência do direito processual.

[4] Essa "tradução" da quarta função semiológica (irrefutabilidade -> irretratabilidade), entre os domínios técnico e jurídico, tenta evitar ambiguidade com outro conceito jurídico, o qual antes negaria ao acusado o direito de sequer postular sua negação de autoria.

validação objetiva (da origem e integridade) de documentos que titulam e transportam chaves públicas, conhecidos como "certificados digitais", os quais podem assim formar encadeamentos conhecidos pelo ambíguo nome de "cadeias de confiança"[6].

Um protocolo de certificação para certificados digitais de chave pública é regido por relações envolvendo entidade certificadora, titulares de pares de chaves assimétricas, documentos eletrônicos contendo dados obrigatórios, e usuários desses documentos. A integridade e a titularidade de uma chave pública é atestada, em documento eletrônico de formato específico conhecido abreviadamente por "certificado digital", mediante validação completa de uma cadeia de confiança (ver nota[6]) nele terminada. Para isso, é essencial que o formato desses documentos represente duas relações: a primeira, entre a chave pública nele contida, e a titularidade do par formado por esta chave pública e sua correspondente chave privada (não contida no certificado), via identificação deste titular; e a segunda, entre uma identificação da entidade certificadora, e uma assinatura digital desta no certificado, que nele representa o ato de sua emissão. Assim, tais certificados servem não só para identificar o titular de um par de chaves cuja chave pública é nele transportada, mas também como meio para distribuição desta chave pública.

Mediante a validação completa de uma correspondente cadeia, os usuários desses certificados podem então inferir que a respectiva chave privada é controlada pelo titular identificado no certificado. Controle este que se presume exclusivo caso a primeira premissa de confiança esteja valendo para esse titular e chave privada, mas premissa esta que nenhum certificado com tal estrutura terá como atestar (como na metáfora aludida inicialmente).

Dos dados contidos em tais certificados, são portanto obrigatórios os seguintes: chave pública, identificação do titular desta chave, identificação da entidade que emitiu

---

[5] Acrônimo de "*what you see is what you get*", referente ao pressuposto de que o que está sendo processado (a nível binário) corresponde fielmente ao que está sendo mostrado na interface de usuário (na tela ou impressora).

[6] Ambíguo no sentido em que a confiança inspirada por tais cadeias, a saber, confiança na titularidade e integridade das chaves contidas nos certificados encadeados, decorre não apenas – como o nome pode dar a entender – do mero encadeamento, isto é, da chave pública em cada certificado na cadeia ser aquela que serve para validar a assinatura no certificado seguinte (e portanto, para validar a titularidade da chave transportada neste). Ela também decorre, e portanto depende também, da validação de todas essas assinaturas e, crucialmente, das três premissas de confiança estarem valendo para a chave pública no certificado que inicia tal cadeia, conhecido por isso como "certificado-raiz".

o certificado, e assinatura digital desta no certificado. E dentre os dados úteis, porém não obrigatórios em formatos-padrão como o X.509 [19], podemos citar os que informam sobre o uso que o titular pretende para este seu par de chaves (no subcampo *userNotice* do campo *policyQualifyer* da extensão *certificatePolicies*), sobre possível revogação antecipada do certificado (na extensão *CRLDistributtionPoints*), dentre outros.

1.2 A equivalência entre físico e virtual

No Brasil, com a assinatura (de punho) da Medida Provisória 2200 pelo Presidente da República em junho de 2001, eventualmente reeditada em versão atualmente vigente – a MP 2200-2 –, foi instituída a Infraestrutura de Chaves Públicas Brasileira (ICP-BR), regime normativo técnico-jurídico ao qual se refere o primeiro parágrafo desta Introdução. Dentre seus dispositivos vigentes, o regime da ICP-BR estabelece que as "cadeias de confiança" sob seu escopo jurídico devem seguir o padrão X.509 com hierarquia única, no sentido de que qualquer certificado autoreferenciado[7] que seja raiz última (ver nota[6]) para tais cadeias, deva ter como titular uma única entidade, nomeada Autoridade Certificadora Raiz da ICP-BR [22]. Conforme o mesmo regime, tal entidade primária é operada pelo Instituto Nacional de Tecnologia da Informação (ITI), que é também responsável pelo credenciamento e descredenciamento das demais certificadoras participantes (cujos certificados podem compor cadeias na ICP-BR), pela supervisão das operações destas, e pela auditoria de processos pertinentes.

Tendo já explicado como uma cadeia de certificados é operada no processo de validação destes[6], resta explicar como é formada no processo de emissão. A Certificadora Raiz, que também é chamada "primária", é responsável pela emissão dos certificados das entidades certificadoras cujos certificados ocupam posição imediatamente inferior nessas cadeias de confiança, e assim por diante. As certificadoras não primárias, também chamadas intermediárias, têm a responsabilidade de emitir, distribuir, renovar, revogar e gerenciar certificados digitais de certificadoras em posição abaixo da sua, ou de clientes finais. Cabe observar que o padrão X.509 estabelece uma espécie de reserva de mercado para certificação, no sentido em que os certificados de clientes finais devem ficar, nas aplicações aderentes ao padrão, desabilitados a verificar assinaturas em outros certificados X.509 (habilitados, portanto, a verificar assinaturas apenas em documentos que não sejam certificados X.509), e os de certificadoras não primárias, passíveis

[7] Frequentemente chamados de "autoassinados", termo ainda mais perigosamente ambíguo por dar a entender que são capazes de atestar sua própria integridade.

de limites para a posição que possam ocupar em cadeias "de confiança" (através da extensão *CertificateBasicConstraints*)

Ainda, para cuidar da coleta e validação de dados de entrada para certificados de clientes finais, dados que irão identificar tais clientes como titulares das respectivas chaves, ou coletar dados para pedidos de revogação antecipada de certificados já emitidos e ainda válidos, e validação de tais pedidos quanto a legitimidade, o regime normativo da ICP-BR estabelece também as chamadas "autoridades de registro", que interfaceiam com as certificadoras intermediárias para tal finalidade. As autoridades de registro devem, ou deveriam, também operar sob credenciamento e supervisão da entidade primária da ICP-BR, ou seja, da Certificadora Raiz operada pelo ITI em sua capacidade fiscalizatória.

1.3 O problema jurídico da função semiológica

Após essa breve introdução aos protocolos de certificação como instrumentos de suporte à distribuição e gestão de chaves públicas, estamos aptos a descrever o problema que será abordado neste trabalho. Tal problema surge na relação entre a função semiológica de irrefutabilidade (no domínio jurídico, irretratabilidade) de assinaturas digitais, e o seu fundamento teórico, que é inviabilidade técnica de se obter, com custo cabível, a chave privada a partir da correspondente chave pública. No domínio técnico, essa inviabilidade – que podemos chamar de premissa de assimetria – é a característica que classifica o correspondente algoritmo criptográfico como assimétrico, caso ela seja inferível para qualquer par de chaves útil ao algoritmo.

Para oferecer garantias dessa inviabilidade, ou seja, garantias de que qualquer chave pública certificada possui tal característica (de assimetria), garantias estas que o escopo jurídico da ICP-BR decreta suficientes, para o Direito Civil, pelo disposto no § 2º do art. 10º da MP 2200-2, o regime da ICP–BR estabelece normas adicionais, de cunho técnico correspondente, com diversas especificações e parâmetros para algoritmos, como por exemplo o RSA (que se tornou "padrão de fato"), softwares e *hardwares* homologáveis, tanto para geração como para uso de pares de chaves sob tal regime.

O problema – peculiar à ICP-BR, quando comparada a outras iniciativas do gênero[8] –, surge da citada presunção de suficiência jurídica, insculpida no § 2º, Art. 10º da MP 2200-2[17]. Se um agente oportunista puder gerar, com custo cabível, a chave privada correspondente a uma chave pública a partir desta, o titular deste par de chaves estará

exposto a sérios riscos e problemas de caráter jurídico, de difícil solução no âmbito da ICP-BR. Então, para que a implícita hipótese de eficácia dessas regras e padrões tenha real valor, é necessário que se analise:

- num primeiro crivo, os métodos conhecidos para se obter chaves privadas a partir da correspondente chave pública;
- num segundo crivo, em mais detalhes, aqueles cuja viabilidade técnica possa ser mensurada por meios empíricos, com distintas distribuições de probabilidade;
- E num terceiro crivo, avaliar se dentre estes existe algum método cuja viabilidade, mensurada em algum sentido prático e sob condições realistas, ponha em cheque a base teórica para tal presunção de suficiência jurídica, sob algum princípio juris doutrinário.

A hipótese de existência de um tal método, e se ele poderia ser usado para robustecer – ao invés de vulnerar – o arcabouço técnico-jurídico da ICP-BR, é aqui tematizada como problema a ser abordado nesse trabalho.

1.4 O teste de Lenstra

No caso do algoritmo-padrão RSA [3], podemos reduzir a verificação dessa hipótese, aqui tematizada, ao exame dos métodos conhecidos para se encontrar os fatores do módulo de um par de chaves – que no caso geral (de módulos regulares) são dois números primos – a partir da chave pública do par. No primeiro crivo, os métodos diretos mais eficientes dentre os atualmente conhecidos – como o de fatoração pelo algoritmo NFS [31], que tem complexidade exponencial na ordem da raiz cúbica do módulo –, são os que tem sido usados para balizar parâmetros técnicos referentes à geração de chaves, em estudos como o de Loebenberger [32] e em normas técnicas como as da ICP-BR aludidas acima.

Porém, a partir de 2012 surgiram novos métodos alternativos, indiretos e potencialmente mais eficientes que os diretos, candidatos ao segundo crivo. Esses métodos se tornaram públicos quando Arjen Lenstra e co-autores os propuseram, em um artigo

[8] Iniciativas de amalgamar regimes técnico (administrativo) e jurídico (algum ramo do Direito vigente) objetivando ordenar a virtualização de práticas sociais que possam fazer uso de criptografia assimétrica, em um regime integrado que geralmente se designa por ICP (ou a correspondente sigla em inglês, PKI – Public Key Infrastructure)

científico publicado em fevereiro daquele ano. O conteúdo desse artigo foi intensamente debatido, entre autores e interessados, num dos principais congressos científicos sobre Criptografia, dois meses depois[9]. Tais métodos empregam técnicas estatísticas para testes empíricos, que na prova de conceito daquele artigo expuseram fragilidades em um dos pilares da terceira premissa de confiança da assinatura digital: a saber, o de que os componentes que geram chaves devem estar sadios.

Essas técnicas assim aplicadas buscam medir a entropia média de geradores de chaves numa amostra de chaves públicas adequadamente dimensionada, colhida dentre as que já foram distribuídas para uso. No caso de chaves RSA, por meio de um cálculo exaustivo porém relativamente simples: o do máximo divisor comum (MDC) entre módulos da amostra. Caso esse teste encontre índice de colisões, na forma de ocorrências de primo comum a mais de um módulo, significativamente maior que índices esperados [31], ou seja, teoricamente estimados pela máxima entropia dos geradores, ou noutras palavras, estimados pela premissa de que os primos selecionados para compor cada módulo devam ser gerados aleatoriamente[10], um tal resultado põe em dúvida a validade da terceira premissa de confiança nos ambientes de origem da amostra testada.

A prova de conceito desta técnica, publicada por Lenstra et. al. em 2012, encontrou índices de colisões de primos muitas ordens de grandeza maior que os índices esperados[11], mas ela tinha duas sérias limitações. A primeira, no fato das características da amostra utilizada terem sido minimamente divulgadas, dificultando a análise das possíveis causas para os índices de colisões encontrados terem sido inesperadamente tão altos. E a segunda, no custo computacional desconhecido, concentrado no algoritmo para cálculo dos MDCs, onde a técnica mais simples conhecida – de varredura dos possíveis pares de módulos na amostra para cálculo direto entre dois módulos – tem complexidade quadrática, o que inviabilizaria (devido ao custo computacional excessivo para se encontrar colisões) essa técnica como base para métodos replicáveis e escaláveis, portanto, qualificáveis ao terceiro crivo da hipótese tematizada.

---

[9] RSA Conference 2012, conforme noticiado em `https://www.networkworld.com/article/2186408/security/alleged-rsa-crypto-flaw-hotly-debated.html`

[10] Mais precisamente, de forma pseudo-aleatória (por se tratar de evento em ambiente computacional determinístico), o que equivale a dizer, com entropia mensurável máxima.

[11] Na amostra inicialmente coletada, com 6.6 milhões de certificados X.509 e chaves PGP contendo módulos RSA, mais de 270 mil (4,3%) compartilhavam módulos, possivelmente entre distintos titulares, vulneráveis assim a ataques de personificação. Dos 12.934 que puderam ser fatorados por terem primo comum com algum outro módulo, afetando 21419 certificados ou chaves PGP, 727 eram de chaves oriundas de cerficados-raiz.

Porém, ainda em 2012, Ilya Mironov, pesquisador da empresa Microsoft, aplicando o método indireto inaugurado por Lenstra, publicou no seu blog "Windows in Theory" um relatório em duas partes descrevendo estudo similar, sobre fatoração de módulos, que ele havia conduzido com seus colaboradores [35]. Esse relatório analisa com mais profundidade a amostra testada em seu estudo, bem como as possíveis causas dos altos índices de colisões também encontrados. E descreve o estado-da-arte para cálculo de MDCs de uma quantidade qualquer de módulos (entre os quais os fatores comuns devem ser raros), baseado em resultados anteriores de D Stehlé & P. Zimmerman, T. Jebelean e D. Bernstein [20], o qual teria sido usado para os testes nesse estudo.

E finalmente, seis meses depois, Alex Halderman, um dos mais importantes e destacados pesquisadores em segurança computacional em atividade, junto com co-autores, publicou um artigo [27] descrevendo outro estudo similar, onde o algoritmo eficiente para cálculo de MDCs descrito por Mironov é implementado, e como esse algoritmo pode compor testes que detectam colisões de primos comuns entre módulos na amostra. A amostra de Halderman, que parece ser a maior até hoje já testada (com cerca de 24 milhões de chaves), foi coletada com ajuda de um *crawler* na Internet, onde foram encontrados índices de colisões de primos também muito acima do esperado, equivalentes aos encontrados no estudo pioneiro de Lenstra e no segundo estudo de Mironov.

Cabe aqui ressaltar o detalhe que distingue o trabalho de Halderman dos dois anteriores, que também empregam métodos indiretos eficazes para se obter chaves privadas de chaves públicas. Em um dos Apêndices do seu artigo, Halderman publicou, com licença livre, o código fonte do algoritmo "FastGCD", de complexidade quasilinear[12], usado para calculo de MDCs entre módulos RSA amostrados para os testes em seu estudo. Com isso, os métodos indiretos para se testar a premissa de assimetria em pares de chaves[13] baseados na técnica proposta por Lenstra, finalmente atingiram nível de qualificação ao terceiro crivo (descrito no fim da seção 1.3) da hipótese aqui tematizada, como pretendemos mostrar ao longo deste trabalho.

---

[12] Ordem de complexidade temporal $O(n(log_{10}n)^2 log_{10} log_{10} n)$[14]

[13] Premissa de inviabilidade, com custo cabível, para se obter a chave privada a partir da correspondente chave pública

1.5 A possibilidade realista de violação da primeira premissa de confiança

Começando pela seguinte observação: com a solução eficiente para teste de colisão de primos disponibilizada por Halderman, atingimos um limiar inédito, na possibilidade estatisticamente significativa (para encontrar colisões), pois proporcional – inclusive em custo computacional – ao tamanho da amostra, de violação da primeira premissa de confiança para assinaturas digitais. Possibilidade realista, ante os índices encontrados em testes já publicados[14], e violação que se propaga às demais premissas: a primeira premissa falha a partir do momento em que um agente A consegue gerar a chave privada de um agente B, quando então o controle de B sobre sua chave privada deixa de ser exclusivo. Como B não tem mais controle exclusivo sobre sua chave privada, não é possível confiar na titularidade de sua chave pública, assim como não se pode presumir que B manifesta ou pode manifestar sua vontade no conteúdo de um documento assinado com sua chave privada. Tudo isso sem que B fique sabendo, se a intenção de A for de fraudá-lo ou prejudicá-lo, com o desafio do ônus da prova pendurado em B pela corrente jurídica do § 2º do art. 10º da MP 2200-2[17].

Como parte desse trabalho, implementamos uma ferramenta de teste, por método indireto, baseada no FastGCD de Halderman, com a qual pretendíamos testar amostras de chaves públicas coletadas de certificados digitais emitidos sob o escopo jurídico da ICP-BR, para avaliar se estão ou não expostas à mesma vulnerabilidade encontrada nos três estudos já citados, em forma de índice de colisões de primos inesperadamente alto. Após implementada a ferramenta, ela foi submetida a uma fase preparatória, em que rodamos um teste preliminar, destinado a avaliar os limites práticos da implementação no ambiente em que foi instalada, e a coletar dados sobre sua performance nesse ambiente. Esse teste preliminar foi executado sobre a maior amostra de chaves públicas que pudemos coletar, contendo as chaves públicas RSA encontradas em um repositório de certificados X.509 disponibilizado pelo projeto SSL Obervatory[15] da ONG Electronic Frontier Foundation (EFF).

Nesse teste preliminar, foi também encontrado índice de colisões de primos inesperadamente alto, também equivalente aos encontrados nos testes publicados por Halderman, Mironov e Lenstra. O que contribui para sustentar a hipótese de vulnerabilidade

---

[14] Índices que variam entre aproximadamente 0.5 e 1.2% de módulos fatoráveis, em amostras que vão de 6 a 24 milhões de chaves RSA, em 3 estudos publicados antes deste.

[15] `https://www.eff.org/observatory`

generalizada – no uso do RSA em larga escala –, de causa ainda especulativa, manifesta em termos de índices de colisões de primos inesperadamente altos e semelhantes em amostragens independentes.

Podemos supor que nossa amostra é ao menos parcialmente independente das de Halderman e de Lenstra pois as chaves públicas que coletamos estavam em certificados quase todos já expirados ao tempo em que o repositório da EFF foi acessado, com data de gravação anterior à da coleta de certificados pelo *crawler* de ambos (em 2012), os quais coletaram certificados que via de regra ainda eram válidos estando em uso na web. Dados colhidos com a execução desse teste preliminar, inclusive sobre limites do ambiente em que a ferramenta foi instalada, e sobre sua performance nesse ambiente durante a execução desse teste, estão registrados em capítulos finais.

1.6 Motivação e caminho percorrido

A intenção inicial com este projeto era a de executar testes, ou disponibilizar a ferramenta para testes, com amostras do repositório das chaves públicas de certificados já emitidos sob o regime da ICP-BR. Em 2015 o repositório completo seria, em tese[16], duas vezes maior que a amostra utilizada na fase preparatória, e três vezes menor do que a amostra testada por Halderman. Porém, não logramos êxito em várias tentativas de obter acesso, seja ao repositório da ICP-BR, ou mesmo a uma parte significativa dele, seja ao efetivo custodiante desse repositório, apesar do que determina o Art. 5° da MP 2200-2[17].

A estratégia de coleta de uma parte significativa desse repositório por *crawling* na Internet não funcionaria para este caso porque a grande maioria dos certificados gerados no regime da ICP-BR se destinam a assinatura digital de documentos, e não a serviços on-line (como por exemplo, via SSL), e portanto, a grande maioria desses certificados não precisa ficar, e por isso não se encontra, on-line. Por uma leitura técnica do conceito de gestão referida nesse Art. 5°, dispositivo que está em uma norma legal equiparável a Lei federal, a responsabilidade pela gerência de todos os certificados emitidos sob o

regime da ICP-BR, caberia à sua Certificadora Raiz, incorporada pelo ITI.

Os termos em que foram dirigidas propostas ao ITI, seja para disponibilização do repositório visando a execução de testes no ambiente em que foi implementada a ferramenta, seja para disponibilização da mesma visando execução de testes em ambiente controlado pelo custodiante do repositório, buscando encontrar uma forma em que a modalidade acordada para testes pudesse ser empregada, junto com providências cabíveis, para robustecer, ao invés de vulnerar, o arcabouço técnico-jurídico da ICP-BR – por exemplo, com revogação "por motivos técnicos" dos certificados que colidirem durante testes –, estão também aqui registrados, em cópias de documentos incluídas como anexos.

Em consequência desta indisponibilidade, numa situação em que sequer tivemos resposta do custodiante legal, nem mesmo para informar se o repositório dos certificados da ICP-BR existe ou não, adaptamos a ferramenta para testes individuais com qualquer chave pública RSA, contra a amostra de módulos contida no repositório que foi nela integrado, e inicializado com os módulos das chaves disponibilizadas no repositório da EFF em certificados já expirados. A ferramenta permite a opção de "prova dos nove", de dedução da correspondente chave privada se o módulo da chave pública testada apresentar colisão com algum módulo do repositório.

---

[16] Número aproximado de certificados então já emitidos, conforme o diretor da Associação de Autoridades de Registro da ICP-BR, Nivaldo Cleto. A declaração citada está disponível no vídeo `https://www.youtube.com/watch?v=L-TWnc2zvBc` , no tempo 28:50, do painel: "Privacidade, Segurança, Criptografia e Identidade digital - Tendências" , no Fórum de Privacidade do CGI-BR, em 2016.

[17] Descrita na seção 7.2 e analisada nas considerações finais

Capítulo 2

Raiz do Problema

2.1 Identificando a raiz do problema

Neste capítulo, vamos averiguar a natureza de uma das vulnerabilidades expostas nos três estudos citados, representada por índice inesperadamente alto de colisões de primos entre módulos RSA, com diagnóstico mais plausível de causa na insuficiência de entropia para o processo de geração de chaves em ambientes de origem da amostra. Tal suspeita principal foi dissecada num desses estudos, o de Mironov, enquanto com Halderman, tais estudos se tornaram replicáveis e escaláveis, a partir da livre disponibilização do algoritmo quasilinear implementado para teste de colisões de primos entre módulos, validando assim também a hipótese tematizada nessa monografia, onde tal vulnerabilidade ganha status de problema para protocolos de certificação digital. Problema técnico de segurança, pois no cenário atual esses protocolos vêm sendo usados, em larga escala, com chaves RSA.

Já no domínio jurídico, o risco decorrente dessa vulnerabilidade (ou de seu correspondente problema de segurança) é o de falha nas premissas de confiança para uso de chaves criptográficas certificadas. Risco que passa a ser bem maior que o teoricamente estimável por avaliações isoladas, mediante a existência de procedimento computacional violador replicável e de eficácia escalável, que ultrapassa em muitas ordens de grandeza essas estimativas, tornando-as irreais. Mais precisamente, mediante técnica indireta de se fatorar o módulo de uma chave pública calculando-se MDCs entre módulos, para daí derivar a chave privada correspondente, caso o módulo desta chave pública venha a colidir com outro de uma coleção, agregada seja para testes, seja para um tal procedimento visando violação dessas premissas.

É necessário então que perguntemos como o método indireto implementado e validado nos três estudos citados foi capaz de expor uma vulnerabilidade que se transforma em problema de segurança para protocolos de certificação. Existe uma grande quantidade de possíveis explicações para a ocorrência de índices inesperadamente altos de colisões de primos nesses estudos, confirmado agora também em nosso teste preliminar, mas neste capítulo abordaremos apenas três dentre as mais prováveis explicações, de natureza técnica, tomando como principal referência o estudo publicado por Mironov [35],

Analisaremos primeiramente a confiabilidade do próprio algoritmo RSA. O tamanho dos módulos utilizados para geração de chaves para esse algoritmo pode variar e essa variação poderia, aparentemente, contribuir para elevar o índice de colisões numa amostra que contenha módulos de tamanhos variados. Em seguida, será analisada a possível relevância da variação de tamanho dos intervalos entre primos consecutivos entre os números inteiros ordenados. E por fim, a questão relacionada à inserção de entropia no processo que seleciona números primos para compor módulos, através do uso de geradores pseudorandômicos (PRNGs).

2.1.1 O RSA é (isoladamente) robusto?

Pode-se culpar o algoritmo criptográfico RSA em si? Como será que a quantidade de primos gerados para compor módulos em eventos independentes e de tamanhos variados, aumenta a probabilidade de um primo se repetir em diferentes módulos, e portanto, de se envolver em colisões? Um módulo RSA é um produto de primos, via de regra dois[1], os quais neste caso são recomendados serem de tamanho igual. De tamanhos iguais, esses dois primos devem ter então metade do tamanho do módulo RSA. O tamanho do módulo é parâmetro independente na geração de chaves. Consequentemente, nas amostras coletadas para os estudos citados aparecem módulos de tamanhos variados. Nessas amostras, a grande maioria dos módulos tinha entre 1024 bits e 2048 bits, e quase sempre em um desses dois tamanhos[2]. Ao passo que, nos

[1] Para módulos ditos "regulares". Cabe aqui observar que, dentre os três estudos citados, apenas o estudo pioneiro de Lenstra mencionou a possibilidade de módulos não regulares ocorrerem na amostra, ao notar que eram regulares todos os módulos aderentes aos padrões para chaves RSA que foram fatorados mediante colisão (conforme nota de rodapé 4 em [30])

[2] No primeiro teste do estudo pioneiro de Lenstra, por exemplo, 73,9% dos módulos tinham 1024 bits, e 21,7% tinham 2048 (mais de 95% com um desses dois tamanhos). Ao passo que, dentre as colisões encontradas, a porcentagem representativa foi maior entre os menores (de 1024 bits).

testes de colisão, a grande maioria dos módulos fatorados tinha 1024 bits, com pelo menos um fator primo de até 512 bits. Avaliando superficialmente, pode-se imaginar que módulos de tamanho pequeno teriam implicação na ocorrência de altos índices de colisão. Então, concentremo-nos por enquanto neles.

A densidade de números primos de 512 bits é aproximadamente $1/ln(2^{512}) \approx 1/350$. O que significa que a probabilidade de se escolher um número primo no intervalo de 1 até 2^{512} é de aproximadamente 0.285% e o número total de primos nesse intervalo é maior do que 2^{503}. Para alcançarmos uma probabilidade de pelo menos 50% de se observar uma colisão entre primos escolhidos, de maneira puramente aleatória, no intervalo entre 2^{502} até 2^{503}, seria necessário (pelo "paradoxo do aniversário"[3]) a geração de 2^{250} números primos, quantidade 10^{65} vezes maior que a quantidade média envolvida nas amostras estudadas (10^{10} módulos), cujo teste pelo MDC produziu dezenas de milhares de colisões. Devido a essa quase incomensurável disparidade, pode-se concluir não só que a diferença entre tamanhos de módulos na amostra, entre 1024 e 2048 bits por exemplo, é irrelevante para a análise desejada, como também é possível concluir que colisões devidas à densidade de números primos no intervalo dos menores pode ser ignorada.

2.1.2 Intervalos entre números primos

Examinemos um algoritmo típico para selecionar números primos:

1. `Gerar um número ímpar randômico` r `de tamanho` t `bits;`
2. `Se` r `for primo, retorna` r `e para;`
3. $r \leftarrow r + 2$, `volte para o passo 2.`

Onde a expressão booleana no passo 2 acima é um teste por um método de Monte-Carlo[4] para detectar números primos, calculando-se números de Jacobi de uma série aleatória entre os resíduos do candidato r (como por exemplo, o teste de Miller-Rabin). No laço externo do algoritmo acima, os números compostos da sequência vão sendo descartados até que se alcance um que passe no teste de primalidade.

---

[3] Princípio de contagem combinatória que ganha o nome de um exemplo simples mas antiintuitivo de sua aplicação . Vide `https://pt.wikipedia.org/wiki/Paradoxo_do_anivers%C3%A1rio`

[4] "Método para construir uma classe de algoritmos que calculam probabilidades baseadas em uma série de amostragens aleatórias,..."([38] Pág. 18)

Considerando os candidatos r aptos a passar no teste por serem primos, sabemos que a distribuição desses não é uniforme, já que depende não só (estatisticamente) da densidade de primos no intervalo delimitado pelo tamanho t escolhido, mas também da distribuição precisa dos primos nesse intervalo. Assim, qualquer primo de tamanho t é selecionável por um algoritmo desse tipo com probabilidade também proporcional ao tamanho do intervalo que o separa do primo antecessor (supondo a inicialização de r puramente aleatória).

Teria essa não uniformidade dos intervalos entre primos algum efeito significativo no índice de colisões de primos entre módulos, encontrados nos testes?

Para uma estimativa desse efeito, consideremos o seguinte: Seja $p_1, p_2, \ldots, p_m$, onde p_m são os primeiros m primos. A probabilidade do i-ésimo número primo ser escolhido será:

$$\frac{(p_i - p_{i-1})}{p_m}.$$

Chamemos essa distribuição de Π. A probabilidade de duas amostras independentes dessa distribuição colidirem, conforme descrita por Mironov, é dada por:

$$Pr_{a,b\leftarrow\Pi}[a = b] = \sum_{i=1}^{m} Pr_{a,b\leftarrow\Pi}[a = p_i].Pr_{a,b\leftarrow\Pi}[b = p_i] = \sum_{i=1}^{m} \frac{(p_i - p_{i-1})^2}{p_m^2}$$

Dada a probabilidade de uma única colisão, o número esperado de colisões entre n amostras independentes e puramente aleatórias sobre Π é $\binom{n}{2}$ $n^2/2$ vezes maior (por aproximação linear). Se os primos estivessem distribuídos uniformemente entre os números inteiros ordenados, então a probabilidade de colisão seria $1/m$ e, para garantir que a quantidade esperada de colisões seja de pelo menos 1, o número de amostras n deve ter tamanho mínimo de $\sqrt{2m}$ (o limite do paradoxo do aniversário[3]). Porém, primos consecutivos não se encontram uniformemente espaçados entre os números inteiros ordenados, e intervalos relativamente grandes ocorrem.

O matemático Atle Selberg provou [18], usando a hipótese de Riemann [21], a melhor cota (condicional) superior temporal conhecida para o calculo da soma

$$\sum_{i=1}^{m} \frac{(p_i - p_{i-1})^2}{p_i} \approx O(\log^3 p_m)$$

o que se traduz no seguinte limite para probabilidade de colisão:

$$Pr_{a,b \leftarrow \amalg}[a = b] = O(\frac{\log^3 p_m}{p_m}) = O(\frac{\log^2 m \log\log^3 m}{m})$$

visto que:

$$p_m \approx m \log m.$$

O limite de Selberg significa que o efeito de espaçamentos irregulares entre primos consecutivos pode aumentar assintoticamente por um fator de ordem

$$O(\log^2 m \log\log^3 m)$$

comparado ao caso simplificado (de espaçamento uniforme entre primos). Assim, examinando essa quota sublinear, é possível concluir que o espaçamento não uniforme entre primos consecutivos é insuficiente para explicar a discrepância em várias ordens de grandeza entre os índices encontrados em testes de colisão nas amostas estudadas, e os índices teoricamente estimados com tal simplificação (que considera intervalos uniformes entre primos consecutivos).

2.1.3 A raíz da vulnerabilidade

Observamos até aqui que a probabilidade de colisão de primos devida ao tamanho da amostra de módulos, ou devida à variação nos tamanhos dos módulos, ou devida ao espaçamento irregular entre dois primos consecutivos dentre os inteiros ordenados, não é causa significativa para o índice de colisões encontrado estar tantas ordens de grandeza acima do esperado, quando se considera as características do RSA como algoritmo isolado. Qual seria então o real motivo para os índices de colisão encontrados com testes de MDC entre módulos RSA em amostras reais?

Resta então considerar a terceira hipótese: a saber, a de que os resultados desses testes reais, comparados aos índices de colisão teoricamente antecipáveis, estão indicando baixa entropia na geração de chaves em ambientes de origem da amostra. O que apontaria falhas de projeto ou de implementação, ou erros de programação, nos softwares que geram as chaves, e não falha matemática em estimativas de índices esperados ou no algoritmo RSA. Ou seja: uma sutil, porém catastrófica, diferença prática real entre o pseudorandômico e o puramente randômico.

Se houver falha sistemática ou aleatoriedade insuficiente na inicialização da fonte de

entropia que, na etapa de geração de chaves, inicia o processo para seleção dos números primos que irão compor o módulo da chave, então duas chamadas independentes a essa fonte poderão resultar em *outputs* com índice inesperadamente alto de bits coincidentes, em cujo caso esses bits de inicialização "randômica" (pseudorandomica) em instâncias "independentes" poderiam propagar coincidências, e convergi-las para eventuais colisões ao longo do processo de geração descrito acima, até a seleção de um mesmo primo.

Capítulo 3

Um problema ou vários?

3.1 Análise dos possíveis problemas

A hipótese de existência de método indireto que pode expor vulnerabilidades no arcabouço técnico-jurídico da ICP-BR foi confirmada pela divulgação dos três estudos estatísticos citados, que utilizam, dentre outros recursos, algoritmo quasilinear para fatoração (via MDC) de módulos de chaves RSA. A parte desses estudos relativa à vulnerabilidade explorável por fatoração de módulos foi validada neste trabalho, pelo teste preliminar realizado como etapa preparatória.

Dentro do que propomos nele tematizar, caberia então, na etapa seguinte deste trabalho, analisar como esse método e seus resultados podem ser utilizados para defesa, para robustecer ao invés de vulnerar, não só o arcabouço normativo da ICP-BR, mas também os protocolos de certificação disseminados na Internet, em ICPs sob outros regimes normativos.

Com respeito à parte validada neste trabalho, se a mais provável causa (inicialmente analisada no capítulo anterior) para índices inesperadamente altos de colisões – que viabilizam ataques probabilísticos por fatoração de módulos de chaves públicas – está na má escolha da fonte de entropia em processos que geram primos para compor módulos de chaves RSA, então deve ser possível rastrear e identificar as bibliotecas ou programas que implementam código ou especificação contendo falhas. Nessa tarefa, o primeiro desafio seria localizar um ou mais geradores de primos "ruins" ou "perigosos" – no sentido de suspeitos de conterem implementações falhas – disponíveis no mercado.

Nesse desafio, a questão inicial a responder é: Seria possível identificar geradores ruins ou perigosos a partir da amostragem, ou seja, examinando-se apenas as chaves e

certificados que as transportam para a amostra nos testes realizados? Sem a colaboração ou participação do responsável legal pela gestão do repositório de certificados da ICP-BR, não foi possível avançar nas etapas seguintes originalmente planejadas para este trabalho. Nem o será em trabalhos futuros, considerando-se o perfil de uso de certificados sob o regime da ICP-BR (descrito no segundo parágrafo da seção 1.6), e o contexto de indisponibilidade, tanto desse repositório quanto da colaboração ou participação do responsável legal por ele, até aqui absolutas.

Quanto às demais vulnerabilidades encontráveis pelo método indireto aqui descrito, há uma outra, de implicações jurídicas em regimes como o da ICP-BR potencialmente mais graves que as decorrentes do risco de fatoração de módulos via MDC, aqui abordada. A saber, a ocorrência de chaves ou de módulos[1] RSA repetidos em mais de um certificado, se os titulares não forem o mesmo. No estudo de Lenstra, por exemplo, cerca de 4,3% dos 6.185.228 certificados X.509 amostrados tinham módulo ou chave pública iguais ao de outro certificado, onde o risco correspondente[2] não pôde ser avaliado no contexto da amostragem, exigindo cautela e discrição adicionais no tratamento e custódia da amostra coletada.

Diante da deliberada e absoluta indisponibilidade, até o momento de conclusão deste trabalho, de uma amostragem significativa para testes no contexto normativo da ICP-BR, cabe então restringir esse capítulo ao que podemos alcançar. Encerrando-o com um resumo abrangente da investigação conduzida em estudos já citados, relativa ao desafio técnico posto pela vulnerabilidade aqui abordada. A saber, o desafio de identificar geradores ruins ou perigosos a partir da amostragem disponível. O resumo abaixo, extraído do estudo de Mironov, é útil no sentido em que poderia nos servir como roteiro ou modelo para o contexto da ICP-BR, ou poderá nos servir em trabalho futuro, em caso de eventual disponibilidade útil.

---

[1] Módulo e expoente da chave pública repetidos entre certificados implica repetição também da respectiva chave privada, em cujo caso o risco para titulares distintos é o de personificação, se um deles vier a saber desta repetição. Repetição apenas do módulo é muito improvável, pois expoentes de chaves públicas RSA não costumam variar muito: nos mais de 6 milhões de certificados X.509 coletados por Lenstra, por exemplo, 98,49% usavam o mesmo expoente (65537), e 99,99% usavam um dentre os nove mais comuns. Se apenas o módulo for repetido, com exponentes para a chave pública distintos, o risco se reduz a apenas o de vazamento quando ambas chaves públicas forem usadas para cifrar uma mesma mensagem.

[2] O risco quando os titulares de chaves ou módulos duplicados não são o mesmo, ou não representarem a mesma entidade, descrito na nota de rodapé anterior.

3.2 Análise do perfil dos primos em módulos fatorados

Os módulos fatorados por Mironov [27] foram classificados em dois grandes grupos: o primeiro grupo, também detectado por Haldeman[3], é formado por módulos gerados por dispositivo de *hardware* fornecido por um único fabricante; enquanto o segundo, é formado por módulos cujos certificados ou fatores primos apresentam pouco em comum, em termos de dados que permitiriam identificar diretamente o gerador. É razoável supor que essas classificações preliminares podem também ter usado metadados de conexões, coletados pelo *crawler* que gerou parte da amostra utilizada nos estudos de Lenstra e de Haldeman, cujos correspondentes não existem no repositório da EFF utilizado em nossa etapa preparatória. O estudo de Mironov [35], que tenta refinar sua classificação preliminar, começa apelidando o grupo menor de "Zoo".

Uma bateria de testes realizada por Mironov, descrita em seu artigo, detectou, dentre várias curiosidades, esta: para cada um dos menores primos ímpares q, só um por cento em média dos fatores dos 3046 módulos no "grupo Zoo" tem resíduo $1 mod q$. O padrão criptográfico ANSI 9.31 especifica primos fortes[4] para compor módulos RSA, mas primos fortes são muito onerosos de se encontrar. Então, muitas implementações permitem que se opte por seleção de primos que atendem parcialmente essa condição, os chamados "primos seguros"[5]"(menos onerosos), ou nenhuma (menos ainda). A seleção pseudoaleatória de primos fortes p não impediria uma distribuição estatística normal de fatores q pequenos[6] para os $p-1$, mas a seleção de primos seguros, sim. Todavia, uma outra característica detectada no "grupo Zoo" contraria a observação acima como explicação: dentre os p que fatoram esses módulos, a maioria não é de primos seguros. Então, o que explica essa quase completa escassez de fatores primos pequenos para os $p-1$ no "grupo Zoo"?

Mironov identifica então essa escassez quase completa de resíduos $1 mod q$ (para $q = 3$, $q = 5$, $q = 7$, etc) entre os fatores dos módulos no "grupo Zoo" como uma espécie de impressão digital (*fingerprint*), ou "traço" específico, do gerador de primos implementado em uma determinada biblioteca criptográfica, cuja utilização é larga-

[3] Conforme descrito na seção 3.2 do estudo publicado por Haldeman.

[4] Um primo p é dito "forte" (strong) se $p-1$ e $p+1$ tiverem ambos pelo menos um fator primo grande.

[5] Um primo p é dito "seguro" (safe) se $p = 2q+1$ onde q também é primo.

[6] Ou seja, que p tenha resíduo $1 mod q$ para $q = 3$, $q = 5$, $q = 7$, etc.

mente disseminada entre aplicações que fazem uso do algoritmo RSA na Internet. E explica, conforme a seção abaixo.

3.2.1 Geração de primos na biblioteca OpenSSL

Para analisar o procedimento que estaria produzindo esse traço específico ao gerar primos, implementado na biblioteca identificada por Mironov como principal suspeita pelas colisões no "grupo Zoo", replicamos dois pseudocódigos que ele inclui para sua análise. O primeiro pseudocódigo, ele afirma corresponder a um "algoritmo perfeitamente razoável" para selecionar primos classificados como seguros, no qual lhe parece estar baseado o segundo, que codifica o supracitado procedimento na respectiva biblioteca do OpenSSL.

1. Gerar um número randômico r de tamanho n (bits);
2. enquanto r ou $(r-1)$ são divisíveis por qualquer $p_2 \cdots p_{2048}$;
3. $r \leftarrow r + 2$;
4. usar o teste de Miller-Rabin [36] para checar se $(r-1)/2$ é primo, senão vá para o passo 1;
5. usar o teste de Miller-Rabin para checar se r é primo, senão vá para o passo 1;
6. retorna r como *output* (primo).

(obs.: a biblioteca OpenSSL contém uma tabela embutida em seu código com os primeiros 2048 números primos, referidos aqui por $p_1 \cdots p_{2048}$)

3.2.2 Geração de primos não necessariamente seguros na biblioteca OpenSSL

Para módulos RSA que não necessitam de primos classificados como seguros, a biblioteca OpenSSL aplica o seguinte procedimento para gerar um número primo:

1. Gerar um número randômico r de tamanho n bits;
2. enquanto r ou $(r-1)$ forem divisíveis por qualquer $p_2 \cdots p_{2048}$;

3. $r \leftarrow r + 2$;

4. `usar o teste de Miller-Rabin para checar se` r `é primo, senão ir para o passo 1;`

5. `retorna` r `como` *output* `(primo).`

Mironov salienta que, neste procedimento, apenas teria sido retirado o terceiro passo do algoritmo anterior. O segundo passo se manteve inalterado[7] (apesar de aparente falta de justificativa aqui), o que explica a distribuição peculiar de fatores primos dos módulos no "grupo Zoo", para os quais são raríssimas as ocorrências de resíduos $1 mod 3, 1 mod 5, \cdots, 1 mod 17863$, mas onde os primos ditos seguros são minoria. Outro detalhe destacado por Mironov como importante, é o de que a seleção peculiar de primos com essa propriedade não representa vulnerabilidade para a respectiva biblioteca, apenas uma espécie de "traço" característico, originado pelo código específico que dela gera tais primos.

Explicando: um primo qualquer de 512 bits terá essa propriedade com probabilidade dada por $\prod_{i=2}^{2048} \frac{p_i - 1}{p_m} \approx 7,5\%$ [35] no caso de primos de 512 bits. Isso quer dizer que o procedimento utilizado para gerar primos não necessariamente seguros na biblioteca OpenSSL computa função simbólica de contradomínio menor do que a coleção de primos que poderiam ser selecionados sem o crivo dessa propriedade. Reduzido, por exemplo, para 7,5% com primos de 512 bits. Todavia, essa redução dos selecionáveis não é suficiente para causar, sozinha, aumento sensível na probabilidade de colisões[8]. Nem para facilitar, heuristicamente, a fatoração de módulos RSA com um algoritmo especial, que seria efetivo apenas para uma parcela irrisória de módulos[9].

Assim, esse traço característico, originado no segundo passo do pseudocódigo referente ao procedimento para gerar primos não necessariamente seguros que se encontra na respectiva biblioteca do OpenSSL, levou Mironov a concluir, com alto grau de certeza, que 96,72% dos módulos no "grupo Zoo" de sua amostra foram gerados por esta biblioteca. Conclusão que, apesar de derivada de um raciocínio probabilístico, não pode ser estendida aos demais módulos que colidiram na sua amostra.

---

[7] Nessa observação, precedida da adjetivação "perfeitamente razoável" ao apresentar o pseudocódigo anterior, Mironov dá a entender (mas sem explicar) que considera o segundo passo justificado no primeiro algoritmo, talvez por razões de performance, mas não no segundo procedimento.

[8] Tendo em vista as probabilidades calculadas na primeira subseção do capítulo anterior.

[9] Por exemplo, para 0.05625% dos módulos de 1024 bits.

Mironov descarta também a mesma origem para os módulos no primeiro grupo, identificado como oriundo de um único produtor de *hardware*. Assim como Haldeman, em sua publicação Mironov não revela a identidade desse produtor. Apenas afirma que, pelas evidências encontradas, ele pôde concluir que o software utilizado para gerar primos são de natureza distinta nos dois grupos. E conclui apontando um grande vilão, responsável no segundo grupo por essa vulnerabilidade exposta com o método indireto inaugurado por Lenstra: a existência de dispositivos (e sistemas) que falham na inicialização adequada de geradores pseudorandômicos.

Voltamos agora um passo, ao desafio sendo abordado nesse capítulo, que é o de rastrear e identificar bibliotecas ou programas que implementam código ou especificação contendo falhas capazes de explicar índices inesperadamente altos de colisões de primos entre módulos amostrados. Podemos refletir sobre o que encontramos, resumido acima, nos três estudos citados. O estudo pioneiro optou por relegar tal desafio ou não divulgar detalhes[10], enquanto os outros dois descrevem uma classificação preliminar, onde os módulos fatorados se dividem em dois grupos, com o grupo maior rastreável a um único fornecedor de dispositivo de *hardware*. Este fornecedor teria sido notificado a respeito, mas sua identificação não é divulgada. E um desses dois estudos, publicado no blog *Windows on theory* em maio de 2012, rastreia o grupo menor, por um traço característico comum entre os primos encontrados nas respectivas fatorações, identificando a origem de quase todos à biblioteca que gera chaves RSA no software OpenSSL.

Esse rastreamento do grupo menor termina então com uma análise sobre a natureza da falha encontrada na biblioteca identificada. Seria uma falha de implementação, e não de especificação do software, apesar da peculiaridade dessa especificação (que permitiiu o rastreamento), em evitar fatores pequenos nos antecessores dos primos selecionáveis, reduzindo assim o escopo para seleção de primos. Com tal falha classificada como de implementação, e não de especificação, o rastreamento da causa de índices inesperadamente altos de colisões de primos entre módulos amostrados prossegue. Agora, nos contextos onde tal biblioteca é empregada, ou seja, nos ambientes computacionais em que ela é integrada e inicializada para gerar chaves RSA, em aplicações que disponibilizam ou utilizam as correspondentes chaves públicas (amostráveis) on-line.

---

[10] Conforme se depreende da nota de rodapé 7 em Lenstra[30]

3.3 Onde e quando o OpenSSL utiliza geradores pseudorandômicos com baixa entropia?

No mesmo estudo, o investigador comenta a prática comum de dispositivos gerarem certificados de chave pública como parte da sequência de inicialização. De fato, e não só: também aplicações que rodam em *background*, os geram durante a sequência de *boot* do sistema onde foram instaladas, quando inicializadas pela primeira vez. Nesses casos, a menos que o gerador pseudorandômico utilizado para alimentar a geração de chaves receba uma semente inicializadora de uma fonte de entropia suficiente, o seu *output* será previsível, de escopo inadequado ou fixo. Em que situações isso ocorre?

Assunto para o capítulo seguinte, onde prosseguiremos com essa análise, examinando geradores pseudorandomicos que podem ser livremente analisados. Especificamente, os geradores pseudorandomicos especificados na arquitetura POSIX que são nativos nos sistemas operacionais mais difundidos do mesmo regime de produção de software onde nasce o OpenSSL, que é o de desenvolvimento colaborativo e licenciamento permissivo (FOSS). Antes, porém, oferecemos uma reflexão sobre essa trilha investigativa, no contexto deste capítulo e do próximo.

Para rastrear com sucesso o traço característico encontrado em fatores primos no grupo menor de sua amostra de módulos com colisões, e mais importante, para publicar sem reservas o que quisesse sobre sua investigação bem sucedida com tal grupo, Mironov precisou contar com um recurso essencial. A saber, a disponibilidade irrestrita do código fonte de uma potencial suspeita. E para seu sucesso, inclusive em convencer-nos, ele contou com o fato da sua principal suspeita dispô-lo: uma biblioteca criptográfica amplamente difudida, produzida e distribuída com licença livre.

Mas quanto ao grupo maior? De origem identificada – presumivelmente com bem mais facilidade – a um único fornecedor, sobre o qual nada nos é dado saber? Nem sobre sua identidade, nem da sua reação à postura ética dos investigadores, que optaram por *responsible disclosure*[11]? E para o nosso sucesso, em aprender como as ICPs devem evoluir após expostas a novas vulnerabilidades com o método indireto, nem que seja por nossas escolhas? Nada. Assim como acerca da monopolista ICP-BR, até aqui, nada. Problema também, mas este, de causa em conflito de interesses no jogo de poder

[11] Modelo para priorizar critérios relativos à divulgação de vulnerabilidades descobertas em softwares e produtos computacionais: ver `https://en.wikipedia.org/wiki/Responsible_disclosure`

que se sobrepõe a questões técnicas. Problema grave, pois confiança na acepção aqui definida não pode ser imposta, seja por poder legislativo, financeiro ou bélico.

Ao encerrar este capítulo com tal reflexão, a intenção é permitir-nos interpretações construtivas para o contexto tematizado neste trabalho. O regime de licenciamento permissivo, que é essencial à trilha investigativa a ser prosseguida no capítulo próximo, permite não apenas uso e redistribuição do software licenciado, mas também seu estudo, também para modificações e adaptações, recompilações e testes realistas com o correspondente código fonte. Inclusive contribuições de correções ao mantenedor, e redistribuições de variantes, onde todos podem aprender com a descoberta de erros e tentativas de corrigir falhas. Prerrogativas vedadas nos produtos, ofuscados por *design*, e aos clientes da empresa para quem trabalha o desbravador dessa trilha aqui (Mironov trabalha para a Microsoft). Sigamos então, na trilha do aprendizado (técnico) possível.

Capítulo 4

Geradores Pseudorandomicos no Kernel LINUX

Números e sequências binárias com propriedades mensuráveis de aleatoriedade desempenham um papel crítico e importante na criptografia computacional moderna. Algoritmos como One-Time-Pad, DES, RC4, RSA etc, e protocolos como os das criptomoedas[1], têm sua robustez diretamente relacionada com a implementação adequada dos geradores de números com tais propriedades [28].

Qualquer definição formal de aleatoriedade, porém, corre o risco de ser entendida como paradoxal[2], razão pela qual as que se pretendem sérias costumam ser dadas em discursos ou contextos filosóficos. Para a Computação, que opera com circuitos eletrônicos e dispositivos programáveis via de regra determinísticos, mas onde o conceito é inevitável e se torna cada vez mais importante, a saída é trabalhar com uma noção menos ambiciosa de aleatoriedade, restrita às suas propriedades diretamente mensuráveis. Renunciando-se, por exemplo, à tentativa de se "cercar" ou buscar garantir propriedades imensuráveis ou de mensuração imprecisa, como a irreprodutibilidade por exemplo.

Circuitos ou dispositivos programáveis concebidos para gerar sequências binárias com propriedades diretamente mensuráveis de aleatoriedade são por isso chamados de geradores de sequências pseudoaleatórias, ou mais frequentemente, geradores de números pseudorandômicos, ou abreviadamente, geradores pseudorandômicos, comumente designados pelo correspondente acrônimo em inglês: PRNGs.

---

[1] Essenciais na implementação de conceitos fundamentais, como por exemplo o de *Proof of Work*

[2] Como definir um padrão para a propriedade de não se seguir nenhum padrão?

O objetivo deste capítulo é prosseguir na trilha investigativa resumida ao final do capítulo anterior, explorando os geradores pseudorandômicos padronizados na arquitetura POSIX[3], disponibilizados em versões de núcleos de sistemas operacionais como o desenvolvido e mantido pela fundação Linux. Especificamente, vamos analisar as funções *random* e *urandom* do kernel Linux, em busca de suas fragilidades operacionais.

4.1 Propriedades de um gerador pseudorandômico "seguro"

Uma lista das características externas de um gerador pseudorandomico ideal pode ser encontrada, por exemplo, em[10]:

1. Um adversário que descubra o estado atual do gerador não pode aprender nada sobre as saídas geradas previamente.
2. Um adversário que descubra o estado atual do gerador não pode aprender nada sobre as saídas futuras desse gerador.
3. Um gerador inicializado com os mesmos tipos de entrada já utilizadas em algum momento, deve produzir uma sequência de bits sem qualquer relação com a sequência produzida anteriormente.

4.1.1 Estrutura dos PRNGs *random* e *urandom* no Linux

A estrutura dos geradores PRNG no Linux é formada por três processos assíncronos[26]. No primeiro processo, o sistema operacional coleta dados variados de eventos no kernel. O segundo processo é definido pela alimentação do repositório de entropia (repositório primário), utilizando uma função de "mistura". O terceiro processo ocorre quando bits em ordem aleatória são solicitados. Nesse processo o *output* é gerado, seguido da atualização do repositório primário. A figura 4.1 descreve o fluxo de bytes no gerador pseudorandômico.

A coleta de entropia é subproduto dos eventos originados no kernel pelo teclado, pelo mouse, por interrupções do sistema e por acessos ao disco. Quando um evento

[3] Padrão de portabilidade e interoperabilidade para sistemas operacionais (ver, por exemplo, `https://pt.wikipedia.org/wiki/POSIX`)

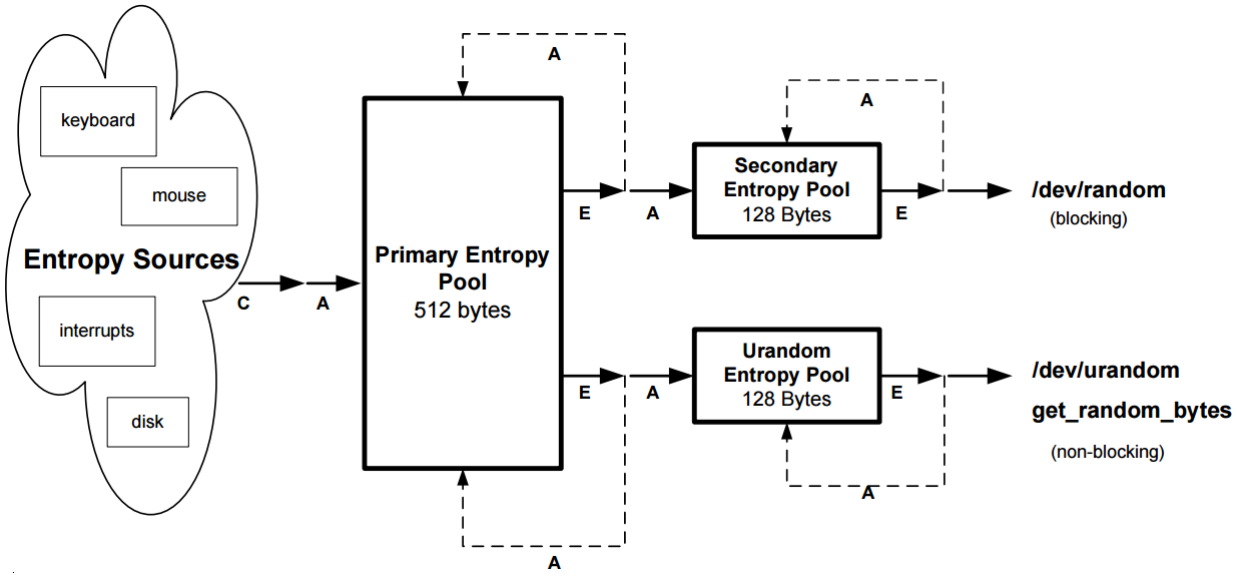

Figura 4.1: esquema PRNG [26]

desses ocorre, duas palavras de 32-bits são usadas como entrada no repositório primário, via processo de mistura. A primeira palavra codifica o tempo medido em *jiffies*[4] ou em ciclos granulares de CPU. A segunda palavra codifica o tipo de evento correspondente.

Como estamos tratando de processos assíncronos para a alimentação dos repositórios, o repositório secundário só recebe dados a partir do momento em que o valor de entropia do repositório primário se torna máximo, isto é, quando o contador de entropia deste atinge o valor 4096. Quando o repositório secundário atinge a sua entropia máxima, o processo no repositório primário é retomado. O processo de adição ao contador de entropia[5] é determinado por um cálculo da estimativa de entropia entrante no respectivo repositório.

4.1.2 *Outputs* dos repositórios

Existem somente três *outputs* gerados pelos repositórios presentes na estrutura dos geradores PRNG no Linux . A extração de bits do repositório primário ocorre somente quando o repositório secundário ou o repositório *urandom* não têm entropia suficiente e precisam ser realimentados. A extração de bits de *urandom* ocorre quando um processo

[4] Unidade que corresponde ao número de milissegundos de diferença no tempo de relógio em que o evento ocorreu, e o tempo em que o sistema foi inicializado.

[5] No sentido de medida de incerteza quanto à previsibilidade habilitada pelo contexto, na forma definida pela Teoria Matemática da Informação de Claude Shannon

de usuário ou do kernel pede uma quantidade qualquer de bits aleatórios pela chamada */dev/urandom* ou *get-random-bytes*, respectivamente. A extração de bits do repositório secundário ocorre quando um usuário usa a chamada */dev/random*.

O processo de extração de entropia (ou seja, de dados "aleatórios") ocorre em três etapas.

1. `Extração de bits "aleatórios" como saída;`
2. `Decremento do contador de entropia do respectivo repositório relativo à saída (na etapa 1);`
3. `Atualização do conteúdo presente no repositório, envolvendo o` ***`hashing`*** `do conteúdo dos repositórios utilizando SHA-1 e a adição dos resultados nos repositórios.`

Cada processo citado acima será abordado com maior detalhamento nas seções posteriores.

A principal diferença entre os PRNG em */dev/random* e em */dev/urandom* está no fato de que, caso o contador de entropia do repositório secundário seja menor do que a quantidade de bits solicitados, a operação de extração de bits é bloqueada e o estado do repositório é realimentado com novos dados extraídos do repositório primário. Enquanto com o*dev/urandom* esse bloqueio não ocorre, e a atualização de entropia no repositório *urandom* só ocorre após o fim de todo o processo de saída de dados.

4.1.3 Adição ao contador de entropia

A quantidade de entropia adicionada a um contador é estimada a partir da diferença de tempo de um evento para outro, sem levar em conta o tipo de evento capturado. Mesmo eventos que adicionam zero a esse contador, têm importância na atualização do estado do gerador. É importante ressaltar que o evento de escrita em disco a partir de um processo de usuário tem valor zero para o cálculo da estimativa de entropia. O contador tem seu valor decrementado em n, quando n bits são extraídos do repositório. Quando a extração é resultado de uma transferência entre repositórios, esse valor n é adicionado ao contador do repositório destinatário.

4.1.4 Atualização dos repositórios

O mecanismo para atualização dos repositórios é baseado em TGFSR (*Twisted Generalized Feedback Shift Register*). A principal vantagem do uso dessa técnica é de prover período (órbita) máximo para qualquer semente inicial não nula, propriedade necessária ao objetivo de entropia máxima do PRNG. O período máximo de uma TGFSR com estado de 128 words pode chegar até $2^{(128*32)} - 1$. Entropia é adicionada a cada atualização de estado ou saída de dados. O algoritmo de *add(pool,j,g)* é utilizado para adição de entropia, onde a variável *pool* é um vetor de bits de 32 ou 128 palavras, j é o índice do vetor, e g é a nova palavra a ser adicionada no repositório.

São utilizados dois polinômios irredutíveis no TGFSR de atualizacão dos repositórios. O polinônio

$$x^{128} + x^{103} + x^{76} + x^{51} + x^{25} + x^{1} + 1$$

é utilizado para realimentar o repositório primário, e o polinômio

$$x^{32} + x^{26} + x^{20} + x^{14} + x^{7} + x^{1} + 1$$

é utilizado para realimentar tanto o repositório secundário quanto o repositório do PRNG *urandom*. As potências que ocorrem nos polinômios indicam os índices dos bits em g que serão operados pela função xor com os bits nos respectivos índices no parâmetro j.

A "adição" de entropia no LRNG (*Linux Random Number Generator*) pode ser entendida como uma atualização na semente a cada interação. O TGFSR pode ser analisado como uma função que encripta o *input* da entropia.. Essa atualização de semente altera as propriedades estruturais dos TGFSR, a ponto de não mais ser possível presumir que o período máximo associado ao respectivo polinômio continue valendo nesse contexto de operação, pois cada processo passa assim a computar uma função simbólica que não mais corresponde a função algébrica linear sobre o estado inicial.

4.1.5 Extração de Bits aleatórios dos repositórios secundário e *urandom*

Extrair bits aleatórios de um repositório não é uma operação trivial [1]. Tal operação envolve o *hashing* dos bits extraídos, modificação do estado atual do repositório, e subtração no correspondente contador de entropia pelo número de bits extraídos. A

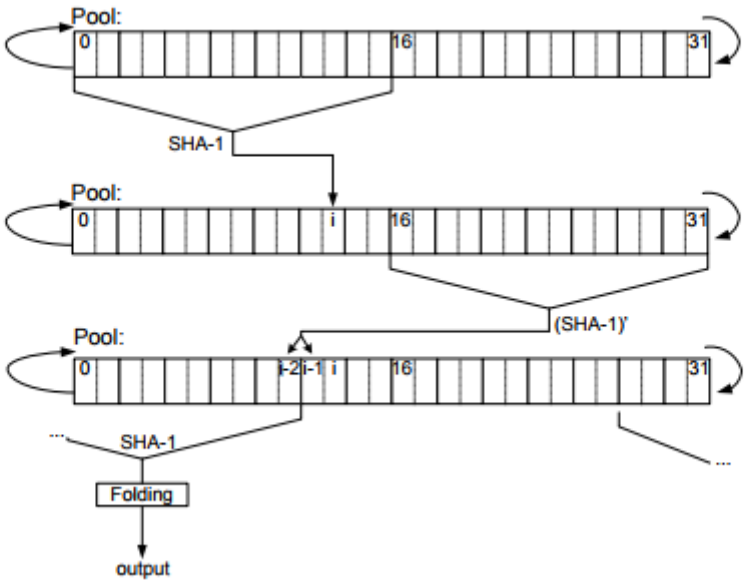

Figura 4.2: Extração de bits aleatórios da função TGFSR [26]

extração de bits de um tal repositório é realizada em blocos de 10 bytes. O algoritmo aplica a função de hash SHA-1 em 16 palavras do repositório, chama a função de adição do TGFSR, na forma: *add(pool, j , SHA-1(pool[0...15]))*, e o valor resultante da função de adição é escrito na posição j. Em seguida, aplica-se uma variante da função SHA-1 – a SHA-1'[6] – na metade direita do repositório e o resultado é adicionado as posições $j-1$ e $j-2$. Por fim, aplica-se SHA-1‘ às 16 palavras anteriores ao $j-2$. A figura 4.2 apresenta um diagrama ilustrativo do referido algortimo de extração.

Os resultados das operações de *hashing* e *folding* são utilizados em uma função de dobra (ou *folding*[7]) que gera um *output* de 10 bytes. O resultado então é colocado em um *buffer* e o processo de realimentação desse *buffer* se repete até o *buffer* atingir a quantidade de bits solicitada pelo processo de usuário ou do kernel. Caso o processo solicitado a fornecer bits aleatórios gere mais bits do que o necessário, esses bits sobressalentes são utilizados no processo de atualização do repositório de onde a extração solicitada ocorreu.

---

[6] A função SHA-1‘ é uma variação de SHA-1, onde as 5 últimas *words* (ou 10 últimos bytes) do repositório são alteradas a cada iteração passada da extração de bits. Essa alteração na função de *hashing* é importante para aumentar o custo de ataques de criptoanálise ao algoritmo

[7] *Input*: W_0, W_1, W_2, W_3, W_4; *Output*: $W_0 \oplus W_3, W_1 \oplus W_4, W_{2_{0...15}} \oplus W_{2_{16...31}}$

4.2 Fragilidades no LRNG

Nesta seção, abordamos formas de ataque ao LRNG descritas na literatura especializada, considerando a premissa de que estimativas da complexidade desses ataques, em termos de custo computacional, servem de aproximação às estimativas empíricas para perda de entropia por falhas correspondentes na inicialização de geradores de primos, descritas no capítulo anterior como principais suspeitas por colisões entre módulos gerados com o OpenSSL, em contextos onde a biblioteca geradora opera sobre plataforma GNU/Linux.

4.2.1 Ataque de criptoanálise na propriedade 1. no LRNG

O primeiro tipo de ataque a ser analisado aqui é o ataque de criptoanálise sobre a propriedade ideal 1. (descrita na Seção 4.1), referente à premissa de imprevisibilidade de um estado passado, a partir de um estado presente que venha a se tornar conhecido pelo atacante. Observamos que o *output* extraído de um repositório é calculado como o último estado do algoritmo de extração. Ou seja, a extração é computada após a atualização do estado deste mesmo repositório. Este fato implica que se o estado de um repositório em um tempo t é conhecido, então é possível descobrir quais dados foram extraídos daquele repositório durante sua última transição de estado (em outras palavras, é possível descobrir o *output* que foi computado na transição do tempo $t-1$ para o tempo t). Dessa forma, qualquer atacante que consiga observar o estado do LRNG em algum instante, pode computar o último *output* do LRNG.

O ataque por criptoanálise tem sua eficácia ligada à realimentação dos repositórios secundário e *urandom*. Caso ocorra a realimentação do repositório, a aleatoriedade dos novos dados dificulta a possibilidade do atacante prever o estado atual do repositório alvo. Dessa forma, o ataque por criptoanálise é melhor empregado contra o repositório *urandom*, que não bloqueia a operação de extração de dados quando o seu contador de entropia atinge valor menor do que a quantidade de bits solicitados.

Caso o ataque se inicie em um instante onde o contador de entropia do repositório secundário esteja alto e a entropia adicionada ao reservatório primário seja previsível, o ataque de criptoanálise também pode ser relevante contra o repositório secundário. Já sabemos que é possível encontrar um $estado(t-1)$ e a saída desse estado a partir de um $estado(t)$ conhecido. De forma análoga, chegamos à conclusão que é possível encontrar os estados: $estado(t-2)$, $estado(t-3)$, ..., $estado(0)$ (onde 0 simboliza

o momento em que a última realimentação do repositório ocorreu) e suas respectivas saídas.

Este fato pode estar associado, em tese, às situações onde estados relativamente próximos nessas sequências, ocorrendo em eventos independentes mas de natureza similar (por exemplo, durante o *boot* de plataformas distintas porém do mesmo modelo e configuração), acabem por induzir dois geradores de primos a eventualmente selecionarem o mesmo primo. Onde saídas distintas do LRNG alimentam instâncias de inicialização nos respectivos geradores, distintas mas que ocorrem no mesmo intervalo entre dois primos consecutivos selecionáveis. Efeito colateral que tende a recorrer, devido à previsibilidade da sequência de estados a partir do momento zero de "um mesmo" evento de *boot* (em plataformas distintas). Efeito que se amplificaria se os intervalos entre primos selecionáveis estiverem despropositadamente "esticados", como caracterizado pela peculiaridade do gerador de primos não necessariamente seguros do OpenSSL descrita por Mironov.

4.2.2 Ataque na geração de Entropia do reservatório primário

Um atacante pode influenciar diretamente a alimentação do repositório secundário. Quando o contador de entropia no repositório primário está cheio, entropia é adicionada diretamente ao repositório secundário. Esse fluxo de coleta de entropia e alimentação direta de um repositório pode possibilitar um atacante a habilidade de criar um estado previamente conhecido, com saída previamente conhecida. Com relação a esse fato, a questão que aqui se coloca é: poderia o mesmo efeito acontecer devido a alguma contingencia na plataforma, então como falha no potencial de entropia esperada[8] do LNRG, de maneira a compor o efeito colateral na geração de primos descrito acima?

4.3 Considerações Finais

Neste Capítulo foi possível, graças ao fato do kernel Linux ter licença livre, observar a complexidade lógica que se apresenta em algoritmos para geração de números pseudorandômicos. Assim como as várias possibilidades de falha operacional, em seus possíveis desenhos, quanto à premissa de imprevisibilidade[9].

[8] Objetivo ideal de um desenho consistente para PRNG, dificilmente traduzível em especificações técnicas.

[9] Mesmo, ou principalmente, quando indiretamente almejada.

Concluímos o Capítulo mencionando a tese de Turing, sobre a qual o mesmo diz ser um pecado acreditar que um software (uma máquina determinística) seja capaz de produzir um número tido como puramente aleatório. E finalmente, com a premissa de que o leitor agora tem conhecimento necessário para entender a vulnerabilidade explorada e explorável pelo algoritmo proposto por Lenstra [30], implementado e distribuído com licença livre por Halderman [27]. Ou pelo menos, uma ideia da profundidade e complexidade do problema abordado neste trabalho.

Capítulo 5

Entendendo o algoritmo implementado por Halderman

5.1 Introdução

Nesta seção, descreveremos o algoritmo estado-da-arte para o método indireto que permite derivar os parâmetros de uma chave privada a partir da chave pública correspondente, mediante fatoração do módulo desse par de chaves. A fatoração se torna viável pelo método indireto que calcula Máximos Divisores Comuns (MDCs), de forma mais simples entre o módulo da chave e outro módulo de uma amostra coletada de chaves públicas. O MDC calculado indica se há ou não fator comum entre módulos envolvidos no cálculo, fato que aqui chamamos colisão.

Um par de chaves pública e privada é gerado para ser utilizado em algum algoritmo de criptografia assimétrica – como por exemplo, o RSA, que atualmente é o mais usado –, e via de regra por algum protocolo que titula e distribui as chaves públicas dos participantes. A confiabilidade desse protocolo depende do sigilo das respectivas chaves privadas, bem como da preservação da integridade das correspondentes chaves públicas tituladas, durante e entre os atos de titulação, distribuição, armazenagem e uso por titulares e terceiros. Os possíveis usos para uma chave pública são: 1) cifragem de mensagem ou documento destinado ao titular da chave pública, ou 2) verificação de assinatura digital em mensagem ou documento supostamente remetida ou assinada pelo titular da chave pública.

Uma chave pública para o RSA é um par (e, N), onde e é a representação binária de um número inteiro que no algoritmo opera como expoente, e N, como módulo para re-

síduos de divisão inteira (donde os nomes). No RSA, N é composto pela multiplicação de números primos distintos, via de regra dois[1] (p e q). Os primos que produzem o módulo N são obtidos em um processo de geração que envolve algum teste de primalidade[2] e seleção de candidatos. Este processo é inicializado por um Gerador Pseudorandômico [11], que por sua vez é alimentado por alguma fonte de entropia fornecida pelo kernel do sistema operacional da plataforma onde a geração ocorre.

No RSA, o mesmo módulo N também compõe a correspondente chave privada (d, N), onde d é outro expoente, também calculado a partir de p e q. O sigilo necessário a d depende do sigilo de p e q. As duas chaves do par tem que ser geradas em conjunto, e os primos p e q, também mantidos em sigilo se for o caso (para maior eficiência no uso da chave privada), ou descartados ao final do processo de geração. Isto porque, se um primo p que fatore o módulo N for encontrado, torna-se trivial obter q, e, com ambos, o expoente secreto da chave privada correspondente, d.

Assim, temos os elementos básicos para descrever o estado-da-arte na implementação do método indireto em exame, que tem custo operacional quase linear. Quando usado para testes de robustez de pares de chaves cujas chaves públicas foram coletadas em uma amostra, o método permite condenar aqueles pares cujo expoente secreto da chave privada pode ser obtido a partir da fatoração de seu módulo, que é público por fazer parte também da correspondente chave pública. Nesses testes, a fatoração de módulos envolvidos em colisão poderá ser obtida por uma mera operação de divisão. Tais colisões são detectadas por cálculos de MDCs, que indicam se há ou não fator comum entre módulos envolvidos. No caso mais simples, se o MDC entre dois módulos for igual a 1, isso indica não haver fator comum entre ambos, e se for maior que 1, indicando colisão, o próprio MDC é o fator comum entre os dois módulos, primo se um dos módulos for regular.

Este método indireto já era conhecido na literatura científica desde 1999, indicado numa resenha publicada por Dan Boneh [15]. Mas o algoritmo mais simples para se calcular MDC entre módulos amostrados, talvez na ocasião presumido também como óbvio ou "o que se tem" para isso, é de complexidade quadrática (de ordem $O(n^2)$) sobre o tamanho da amostra. O custo quadrático decorre da solução óbvia para se testar todas as colisões possíveis, calculando-se diretamente o MDC entre cada dois módulos, percorrendo-se todos os pares de módulos na amostra para isso, pela qual

---

[1] A frequência de Ns regulares foi abordada na nota de rodapé 1 do capítulo 2

[2] Nas análises descritas em capítulos anteriores, o teste empregado é o de Miller-Rabin (mais detalhes em `https://pt.wikipedia.org/wiki/Teste_de_primalidade_de_Miller-Rabin`)

esse algoritmo óbvio tem alcance não escalável.

O custo quadrático sobre amostras de tamanho significativo para uma expectativa razoável de ocorrência de alguma colisão, inviabilizava a descoberta de chaves públicas cujos módulos colidem. Ou mesmo, tentativas: de fato, até onde sabemos, nenhuma quebra ou comprometimento de chaves privadas, já usadas ou em uso, foi divulgado nos doze anos seguintes à publicação da resenha de Boneh[3]. Apesar de já se ter conhecimento, a partir de 2005 com Daniel Bernstein [13], de um algoritmo subquadrático para encontrar fatores comuns numa amostra de números inteiros densa em coprimos. A presunção de que o algoritmo óbvio para cálculo de MDCs era o que se tinha, como parâmetro de custo e de escalabilidade para esse método indireto, só foi definitivamente vencida em 2012, com a publicação do estudo pioneiro de Lenstra e co-autores, pela sua importância e alcance.

Esse estudo foi pioneiro tanto em escala quanto na descoberta de colisões, encontradas às dezenas de milhares numa amostra com mais de dez milhões de chaves públicas (quase todas RSA) já utilizadas ou em uso na Internet. E importante pelo fato dos índices de colisão encontrados superarem, em muitas ordens de grandeza, qualquer expectativa plausível que as teorias e conceitos pertinentes pudessem até então indicar, como vimos nos dois capítulos anteriores. A pista de como essa escalada pioneira foi possível, estava lacônica na sétima nota de rodapé do artigo de Lenstra: *Sapientis sat.* Então, em maio do mesmo ano, Mironov publica estudo semelhante onde mostra o "pulo do gato" de Bernstein aplicado ao método indireto. E em agosto, Halderman publica o seu, agora com esse "pulo" (para baixo do custo quadrático) não só implementado e testado com sua amostra (até agora a maior), mas também com o respectivo código fonte disponibilizado sob licença livre.

Com a domesticação do "gato que pula[4]" por Halderman, em programa que ele chamou de *FastGCD*, podemos testar na prática seus parâmetros de escalabilidade, para dimensionarmos limites ao tamanho das amostras testáveis no ambiente instalado.

---

[3] Em 2015, o aluno João Henrique G. Sousa defendeu e publicou um trabalho final de graduação na UnB em que usava o algoritmo óbvio indicado por Boneh, em 10 subgrupos da mesma amostra que foi preliminarmente testada neste trabalho. Naquela ocasião, não foi possível calcular o MDC entre todos os pares de módulos da amostra devido a problemas de escala e limitações de performance na plataforma utilizada. Por isso não foram encontradas colisões, que no teste completo com a mesma amostra surgiram: 1152 colisões ao todo.

Mas antes disso, cabe explicar o algoritmo, que é composto de três fases:

As três fases do *FastGCD*, executando sobre uma amostra de $N_i \cdots N_m$ módulos, são:

1. `Criar uma árvore binária de produtos, multiplicando-se pares de nós para instanciar o nó ascendente, começando com módulos` N_i `nas folhas e raiz` $N = \prod_{i=1}^{m} N_i$;
2. `Criar uma árvore binária de resíduos, com a raiz` N `dividida pelo quadrado de cada nó descendente na árvore de produtos, para instanciar o nó em posição equivalente na árvore de resíduos, terminando com` $R_i = N_i^2 mod N$ `nas folhas;`
3. `Calcular` G_i `= MDC(`R_i^2`,` N_i`), para cada índice` i `da amostra de módulos.`

Esse algoritmo pode então ser aplicado para encontrar colisões entre módulos distintos de uma amostra, sem precisar calcular diretamente o MDC entre dois que colidem, e para fornecer a fatoração de praticamente todos que colidirem na amostra. As funções assintóticas que estimam o custo computacional desse algoritmo são de ordem $O(mnlog(m)log(mn)loglog(mn))$ para a árvore de produtos e para a árvore de resíduos, e de ordem $O(mn(logn)^2 loglog(n))$ para o cálculo dos MDCs na terceira fase, perfazendo o total estimado por estas cotas de complexidade, em custo quasilinear.

5.1.1 Árvore de Multiplicação

Como já explicado acima, a primeira fase do algoritmo consiste em construir uma árvore binária contendo, em cada nó, o produto dos dois números inteiros instanciados nos descendentes imediatos, começando com as folhas instanciadas pelos módulos N_i da amostra (figura 5.1). Como o nome mesmo sugere, utiliza-se um algoritmo de multiplicação rápida [14, 12] para o cálculo desses produtos (onde o tamanho médio dos operandos dobra a cada nível que se ascende), e como entrada, um repositório de chaves públicas RSA, do qual se extrai a corresponde amostra de distintos módulos N_i.

[4] O caminho para a quase linearidade do algoritmo de Bernstein, que cobre até sua aplicação ao método indireto para fatoração de módulos N_i envolvidos em colisão, foi aberto pela observação da igualdade MDC(N_i^2, N) = MDC(N_i^2, $N mod N_i^2$).

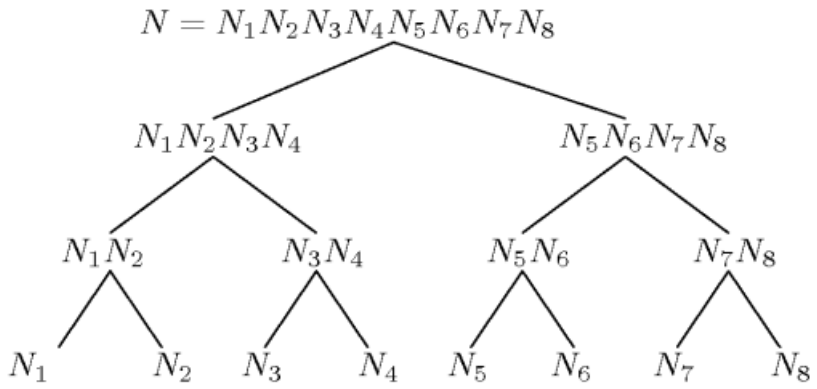

Figura 5.1: Árvore de multiplicação [34] dos módulos para fatoração

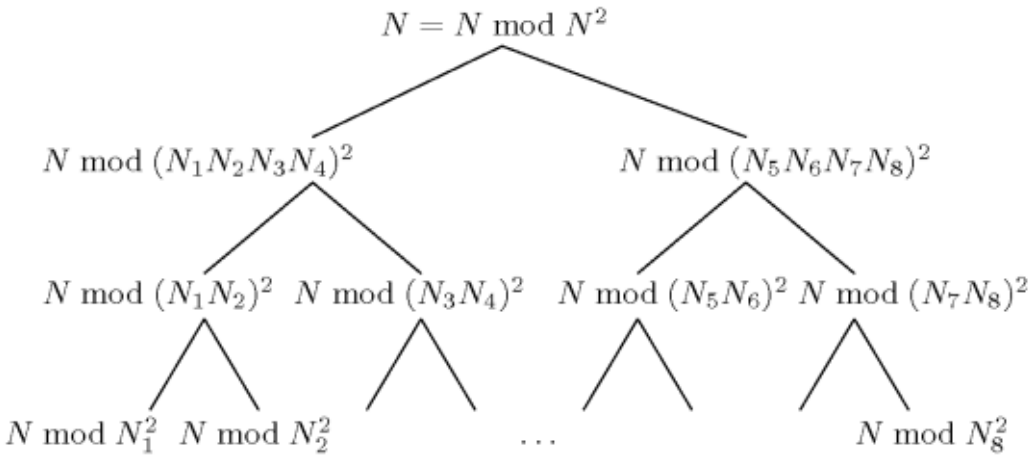

Figura 5.2: Árvore de restos [34] dos módulos utilizada para encontrar a existência de um divisor comum a dois módulos distintos

5.1.2 Árvore de Restos

A segunda fase do algoritmo consiste em construir uma árvore de resíduos para dela se obter a série MDC(N_i^2,N) com o menor custo possível. Obter essa série por cálculo direto dos MDCs teria custo quadrático no tamanho da amostra, equivalente ao de se calcular diretamente os MDC(N_i, N_j) para i, j em $1 \ldots m$. Mas para a série desejada, é possível reduzir significativamente esse custo utilizando-se a árvore criada na primeira fase: podemos percorrer a árvore de produtos da raiz às folhas calculando cada resíduo $N^2 mod P$, onde $P = \prod_{i=j}^{k} N_i$ é a instância no nó cujas folhas descendentes são $j \ldots k$. (figura 5.2)

O resíduo calculado instancia o nó que na árvore de resíduos está na mesma posição

que P ocupa na árvore de produtos. Fazendo isso em ordem descendente, cada cálculo pode ser simplificado pela identidade $N^2 mod P = P'^2 mod P$, onde P' está na posição ascendente imediata à de P.

5.1.3 Máximo Divisor Comum

A terceira fase do algoritmo consiste em percorrer as folhas da árvore de resíduos calculando a série $G_i = \text{MDC}(N_i^2, N mod N_i^2)$. Para detectar colisões e obter as consequentes fatorações, a interpretação do resultado do algoritmo deve ser a seguinte:

Se $G_i = N_i$, então N_i não tem fator comum com nenhum outro módulo da amostra. Caso contrário, $G_i > N_i$ indica ocorrência de fator comum entre N_i e um ou mais outros módulos da amostra. Se $N_i^2 > G_i > N_i$, esse fator comum será próprio, primo (supondo N_i regular), e dado por G_i/N_i. Senão, $N_i^2 = G_i$ indica que N_i tem todos seus primos compartilhados com dois ou mais outros módulos. Apenas nesse último caso[5] a fatoração de N_i requer cálculo direto dos $\text{MDC}(N_i, N_j)$ por varredura de j em $1 \ldots m$.

[5] No estudo de Mironov, em que foram encontradas mais de dez mil colisões, apenas 36 estavam nesse último caso, requerendo calculo direto dos $\text{MDC}(N_i, N_j)$ para j em $1, \ldots, m$

Capítulo 6

Metodologia

Neste capítulo será apresentada a ferramenta desenvolvida para execução de testes individuais com qualquer chave pública RSA, chamada de *Chave Fraca GUI*. Como dito no capítulo introdutório deste documento, a ferramenta tem como objetivo disponibilizar uma interface de fácil entendimento e operação por usuários comuns. Com ela, esperamos que mesmo aqueles sem conhecimento necessário para utilização do complexo algoritmo (*FastGCD*), de Halderman [27], não fiquem reféns da "boa fé" exigida pelo regime da ICP-Brasil, podendo pleitear revogação ou *recall* do seu certificado caso o módulo de sua chave RSA se envolva em colisão com algum outro de uma amostra significativa.

A intenção inicial deste projeto (também citada no capítulo introdutório), que era de viabilizar e/ou executar o teste de Lenstra com o repositório de chaves públicas RSA certificadas sob o regime da ICP-BR, foi perseguida até o limite de prazo para defesa desse trabalho de graduação, mas não pôde ser alcançada pela desfavorável combinação de dois fatores, dos quais o segundo era imprevisível: a saber, uma peculiaridade utilitária da ICP-BR (onde a quase totalidade dos certificados emitidos não precisa ficar disponível *on line*[1]), e o desinteresse do custodiante legal do respectivo repositório (o ITI), que se omitiu em várias tentativas. Nessas circunstâncias, o objetivo desse trabalho foi então adapatado, no limite da prudência relativa a prazos, para produzir a ferramenta *Chave Fraca GUI*, aproveitando-se as implementações, o ambiente e os resultados dos testes preliminares conduzidos em preparação para o objetivo inicial.

Este capítulo é dividido em 4 seções. A primeira seção é destinada a descrever o *setup* do ambiente computacional para o uso do *fastGCD*, componente essencial para

[1] Pelo motivo explicado na introdução (segundo parágrafo da Seção 1.6)

o objetivo inicial. Na segunda seção serão apresentados os resultados obtidos a partir da presente instalação do *FastGCD* com a base de dados disponibilizada pela Electronic Frontier Foundation[2]. Em seguida, iremos apresentar a motivação para criação da ferramenta *Chave Fraca GUI*. Por fim, serão expostas todas as funcionalidades da ferramenta desenvolvida neste trabalho de graduação, de forma semelhante a um "Manual do Usuário".

6.1 O *setup* para o *FastGCD*

O *setup* para instalação e execução do algoritmo *FastGCD* implementado por Halderman é composto por três peças, todas de importância vital. Como o problema da fatoração de um módulo regular em seu par de primos envolve números da ordem de pelo menos 512 bytes, faz-se necessária uma biblioteca para aritmética de precisão arbitrária. A biblioteca livre utilizada pelo *FastGCD* é a GMP[3]. O segundo componente do *setup* é um sistema operacional do tipo UNIX (GNU/Linux). Devido às restrições de manipulação de *threads*, impostas pelos sistemas operacionais MacOS e Windows, apenas um sistema operacional do tipo UNIX alcança a eficiência computacional máxima do algoritmo de cálculo indireto do MDCs sobre um conjunto de módulos RSA. O último elemento envolvido no *setup* é a base de módulos RSA que será utilizada como amostra, para encontrar todas as chaves públicas vulneráveis nesse conjunto de módulos, causada pela ocorrência de colisão (repetição) de fator primo comum com algum outro módulo da amostra.

Realizada as adequações descritas acima, seguimos o *setup* com o *download* do código fonte do *fastGCD* obtido da correspondente referência no citado artigo do autor[4]. A instalação do software é feita a partir da linha de comando *./install.sh*, executada na pasta onde o conteúdo do código fonte foi extraído. É importante ressaltar que o software disponibilizado por Halderman tem licença livre. Para rodar uma nova instância da aplicação, basta usar a linha de comando *./fastgcd input.moduli*, onde *input.moduli* é um arquivo contendo módulos RSA únicos, em hexadecimal, separados por quebra de linha. As subseções abaixo irão revelar alguns detalhes sobre os elementos citados no parágrafo anterior.

---

[2] `https://www.eff.org/observatory`

[3] `https://gmplib.org/`

[4] Disponível em `https://factorable.net/resources.html` e baixado em 07/09/2016

6.1.1 Biblioteca GMP

GMP é uma biblioteca livre para aritmética de precisão arbitrária, operando em inteiros com sinal, números racionais e números de ponto flutuante. O único limite prático para a precisão dessa biblioteca é a memória disponível na máquina em que a biblioteca opera. A biblioteca GMP dispõe de uma rica quantidade de funções e tem o objetivo de disponibilizar uma coleção de funções aritméticas com excelente desempenho computacional para todas as aplicações que necessitam de uma precisão maior do que a suportada pelos tipos básicos da linguagem C na plataforma de hardware subjacente.

As plataformas alvo da GMP são: Sistemas tipo UNIX (GNU/Linux), Solaris, HP-UX, Mac OS X/Darwin, BSD, AIX, etc. Também é compatível com Windows 32-bits e 64-bits. A velocidade da biblioteca é alcançada com o uso da aritmética de tipo *fullword*, com o uso de algoritmos sofisticados, que otimizam o código assembly para os *loops* mais comuns nas diferentes CPUs, e por enfatizar a velocidade de processamento em contraste com a simplicidade e elegância.

Os três tipos de variáveis mais utilizados nesta biblioteca são: mpz_t (inteiro de precisão múltipla), mpq_t (fração de precisão múltipla) e mpf_t (mantissa de precisão arbitrária com precisão de expoente limitado). Neste projeto utilizamos somente o tipo mpz_t para a tipificação das variáveis que compõem um par de chaves RSA. O tipo mpz_t da biblioteca GMP é pequeno, contendo somente alguns tamanhos pré-definidos e ponteiros para dados alocados. Assim que uma variável é iniciada ($mpz\_init(var)$), a GMP é responsável pela alocação de espaço para aquela variável. A partir do momento que uma variável excede o espaço alocado, a biblioteca "aloca" mais espaço em memória. Variáveis do tipo mpz_t e mpq_t nunca reduzem o tamanho do seu espaço alocado. Essa política evita realocações frequentes.

O tipo mpz_t é implementado como um vetor de um só elemento de uma determinada estrutura. Sendo assim, a declaração de uma variável cria um objeto com todos os campos que a GMP precisa para interpretá-lo. É importante ressaltar que os campos gerados para cada mpz_t são exclusivamente para o uso interno da biblioteca e não devem ser acessados diretamente por um código que espera ser compatível com versões futuras da GMP. Todo o processo de alocação de memória é feito mediante chamada à primitiva *malloc()* por padrão. Todas as declarações necessárias para o uso da GMP estão contidas no arquivo gmp.h e foram desenvolvidas para funcionar com compiladores de C e C++ ($\#include < gmp.h >$). Existem cerca de 150 funções nativas da

biblioteca GMP para a classe das funções de aritmética de inteiros com sinal. Essas funções são facilmente identificadas, pois são precedidas de *mpz_* e o tipo associado dessas funções é o *mpz_t*.

6.1.2 Base de módulos RSA

O arquivo base disponibilizado pela EFF é um banco de certificados no formato .sql. Assim, é necessário iniciar um servidor MySQL para extrair os módulos RSA desejados e realizar consultas acerca dos respectivos expoentes. O MySQL usa uma arquitetura cliente/servidor para comunicação, de modo que antes de utilizar o programa cliente (monitor para realização de consultas), é necessário executar o servidor (em geral, chamado *mysqld*). O script SQL da EFF não cria uma base de dados de forma automática, portanto, o primeiro passo é criar uma base de dados SQL nova para receber os dados do repositório da EFF com o comando de linha *CREATE DATABASE database_name;*, sendo *database_name* o nome da nova base de dados. O próximo passo é selecionar a base de dados criada *USE database_name;*. Note que o novo banco criado estará sem nenhuma tabela ou dados, sendo necessário carregá-lo com a base de dados serializados recebidos via download da EFF. Neste passo basta executar o seguinte comando: *SOURCE /path/to/file/observatory-dec-2010.sql*, onde */path/to/file* é o caminho absoluto ou relativo até o script da EFF. A partir deste ponto, supondo que o usuário MySQL corrente tenha os privilégios necessários, basta esperar o script criar a base de dados no ambiente do *setup*. Devido ao extensivo tamanho do script, é necessário um mínimo de 15GiB disponíveis em disco.

Agora que temos acesso a base de dados, podemos extrair os módulos RSA que serão utilizados no arquivo de input do *fastGCD*. Como queremos extrair a maior quantidade possível de módulos dos certificados, não iremos filtrar certificados já expirados, pois afinal, a colisão de um módulo de chave válida com um módulo de chave já expirada já é suficiente para recuperarmos a chave privada do par de chaves ainda válido. A primeira *query* a ser feita é a seguinte: *SELECT REPLACE('RSA Public Key:Modulus', ':', '') INTO OUTFILE '/tmp/rsa_moduli' FROM all_certs WHERE 'RSA Public Key:Modulus' IS NOT NULL;*, de fato a *query* está adicionando ao arquivo *rsa_moduli* todos os módulos não nulos de todos os certificados presentes na base de dados. Com este processo realizado, já temos todo os dados necessários para executarmos o *fastGCD* sobre a amostra de chaves públicas RSA disponibilizada pela EFF.

6.2 Resultados do *fastGCD*

Como já explicado, a motivação inicial deste projeto era testar amostras de chaves públicas coletadas de certificados digitais emitidos sob o escopo jurídico da ICP-BR, para avaliar se estão ou não expostas à mesma vulnerabilidade encontrada nos três estudos já citados [20] [31] [27]. Devido a falta de cooperação do ITI, assunto que será abordado com as devidas ressalvas no próximo capítulo, não logramos êxito em várias tentativas de obter acesso, seja ao repositório da ICP-BR, ou mesmo a uma parte significativa dele[5].

Apesar das adversidades relacionadas à postura do órgão gestor dos certificados emitidos conforme regime da ICP-Brasil, não poderíamos deixar de testar o poder computacional do algoritmo disponibilizado por Halderman, e seus limites operacionais no ambiente computacional disponível para este trabalho. Apresentamos então, a seguir, alguns resultados alcançados durante a fase preparatória de teste, em um computador com as seguintes especificações:

1. Modelo: ASUS ROG G74SX
2. Processador: Intel® Core™ i7-2670QM CPU @ 2.20GHz x 8
3. Memória Ram: 11,7 GiB DDR3
4. Sistema Operacional: ubuntu 15.10 64-bit
5. Disco: 270GB

6.2.1 Coleta de módulos no repositório da EFF

Na seção anterior foi descrito o método utilizado para extrair os módulos dos certificados presentes na base de dados amostrados, e agora serão expostos os resultados oriundos de sua execução no ambiente computaiconal disponível.

1. Inicialmente foram extraídos 4019595 módulos, em 1 minuto e 46,91 segundos

[5] Como mencionado no capítulo introdutório, a estratégia de coleta por *crawling* na Internet não funcionaria para este caso porque a grande maioria dos certificados gerados no regime da ICP-BR se destinam a assinatura digital de documentos, e não a serviços on-line (como por exemplo, via SSL), e portanto, a grande maioria desses certificados não precisa estar, e não se encontra, on-line.

2. Como o algoritmo necessita de módulos distintos entre si, também foram extraídos[6] 3933365 módulos únicos, em 1 minuto e 45,15 segundos

6.2.2 Colisões, tempos de execução e erros

Como mencionado anteriormente, o *fastGCD* têm eficiência proporcional ao ambiente em que ele executa. Sendo assim, obtivemos os seguintes resultados, para as seguintes situações.:

1. 3933365 módulos RSA distintos com apenas um núcleo ativo:
   (a) 457.762s (≈ 7m 36s) para o processamento da árvore binária de produtos.
   (b) 5909.781s (≈ 98m 30s) para o processamento da árvore binária de restos.
   (c) 6508 módulos vulneráveis (0,165% de módulos vulneráveis nessa amostra).
2. 3933365 módulos RSA distintos com dois núcleos ativos:
   (a) 385.129s (≈ 6m 25s) para o processamento da árvore binária de produtos, um ganho de 15,8% em comparação ao teste com somente um núcleo.
   (b) 3309.604s (≈ 55m 9s) para o processamento da árvore binária de restos, um ganho de 43,9% em comparação ao teste com somente um núcleo.
   (c) 6508 módulos vulneráveis (0,165% de módulos vulneráveis nessa amostra, assim como no teste com somente um núcleo).
3. 3933365 módulos RSA distintos com quatro núcleos ativos:
   (a) 309.908s (≈ 5m 9s) para o processamento da árvore binária de produtos, um ganho de 19,5% em comparação ao teste com dois núcleos.
   (b) Após iniciar o processo modular na árvore binária de resíduos, a memória RAM se torna insuficiente para aguentar a demanda exigida pelo algoritmo, ocasionando um erro e um *HALT* geral no kernel do SO.)
4. 4019595 módulos RSA com dois núcleos ativos
   (a) UNIQ[7] levou 3.582s e computou 3933365 elementos

[6] *SELECT DISTINCT REPLACE('RSA Public Key:Modulus', ':', '') INTO OUTFILE '/tmp/rsa_moduli' FROM all_certs WHERE 'RSA Public Key:Modulus' IS NOT NULL;*

(b) 385.129s ($\approx$ 6m 25s) para o processamento da árvore binária de produtos.

(c) 3309.604s ($\approx$ 55m 9s) para o processamento da árvore binária de restos.

(d) 6508 módulos vulneráveis (0,165% de módulos vulneráveis nessa amostra, assim como revelado nos testes com um e dois núcleos).

6.3 A motivação para a ferramenta *Chave Fraca GUI*

Observada a validação teórica da hipótese de existência de um método que derrota a premissa de inviabilidade técnica de se obter, com custo cabível, a chave privada a partir da correspondente chave pública, e visto com esse teste preparatório a comprovação prática desta mesma hipótese, com resultados compatíveis aos encontrados em estudos semelhantes anteriores sobre amostras independentes, tornou-se oportuno a criação de uma ferramenta que auxilie o cidadão brasileiro que se encontra sob o jugo da MP-2200-2. Com o auxílio de um design simples e informativo, a *Chave Fraca GUI* realiza testes individuais com qualquer chave pública RSA, contra a amostra de módulos contida no repositório[8] que foi nela integrado.

6.4 Manual *Chave Fraca GUI*

Esta seção é destinada para o auxílio ao usuário da ferramenta *Chave Fraca GUI*. A ferramenta é composta de quatro aplicações básicas, sendo elas: Teste de robustez da chave pública RSA; Cálculo da chave privada, em caso de colisão detectada; Encriptar mensagem com algortimo RSA; Decriptar mensagem com algoritmo RSA (usando a chave privada calculada em caso de colisão);

---

[7] Ao analisarmos o código fonte de *fastGCD*, foi possível observar a implementação de uma função chama UNIQ. Essa função recebe um vetor de variáveis de tipo mpz_t contendo todos os módulos das chaves públicas e devolve um vetor de mesmo formato, porém, sem nenhum módulo repetido. O algoritmo *FastGCD* tem como entrada um arquivo contendo uma coleção de módulos RSA, em formato hexadecimal, separados por quebra de linha. Para o processamento correto destes módulos existe uma etapa de pré-processamento onde todos os módulos são transformados em binário e os primeiros bytes são responsáveis por armazenar o número de módulos contidos na coleção. A chamada para da função UNIQ é realizada imediatamente após o pré-processamento da entrada.

[8] O repositório foi inicializado com os módulos das chaves RSA extraídas do repositório da EFF.

Teste de robustez da chave pública RSA [9] Nesta opção, o usuário poderá realizar o teste de robustez de sua chave pública. A interface gráfica irá apresentar duas formas de realização do teste, conforme a figura 6.1.

- O usuário seleciona um arquivo que contenha o módulo da sua chave pública em formato hexadecimal e o expoente daquela chave, separados por quebra de linha.
- O usuário digita o módulo da sua chave pública em formato hexadecimal e o seu respectivo expoente, em seus respectivos campos.

Após observar que ambos os campos foram preenchidos, a ferramenta *Chave Fraca GUI* libera o botão para realizar o teste de robustez da chave pública do usuário. Como resultado, a ferramenta informa quanto tempo de execução foi necessário para realizar o teste, informa se encontrou ou não uma colisão e, caso tenha encontrado, informa qual módulo, presente na base de dados, compartilha um primo comum com o módulo da chave pública do usuário. A partir deste momento, o usuário poderá notar que o arquivo: *colisao.input* e *report.tmp* foram criados, ou modificados, na pasta de instalação da ferramenta. No arquivo *report.tmp*(6.2) serão encontradas todas as informações que foram obtidas no teste de robustez. O arquivo *colisao.input* é composto pelo módulo da chave pública testada, o expoente desta chave pública e o módulo que compartilha um primo comum com o modulo da chave testada.

Cálculo da chave privada [10] Caso o usuário obtenha colisão[11] no teste de robustez, a ferramenta *Chave Fraca GUI* dispõe de uma opção para o cálculo da Chave Privada do usuário. A interface gráfica irá apresentar os dados referentes a chave pública do usuário e o módulo colidido para realização do cálculo do expoente da chave privada, conforme a figura 6.3.

A ferramenta *Chave Fraca GUI* realiza a operação do cálculo da chave privada após o usuário apertar o botão designado. Como resultado, é informado o expoente da chave privada que foi calculada pelo programa. O usuário poderá notar que o arquivo: *private.tmp* foi criado, ou modificado, na pasta de instalação da ferramenta. O arquivo *private.tmp* contém o expoente da chave privada calculada pelo programa.

---

[9] O código fonte da aplicação pode ser encontrado no apêndice A, página: 76.

[10] O código fonte da aplicação pode ser encontrado no apêndice B, página: 80.

[11] Encontre uma colisão entre o módulo da sua chave pública com um módulo da base de dados.

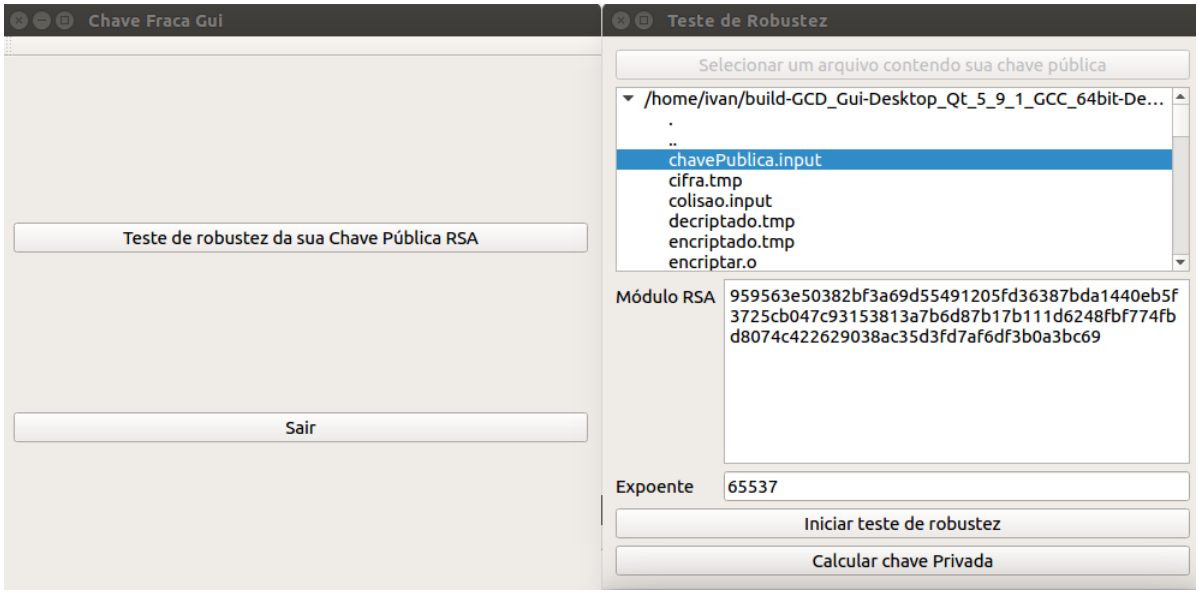

Figura 6.1: Teste de colisão entre Chave Pública e coleção de módulos RSA em hexadecimal

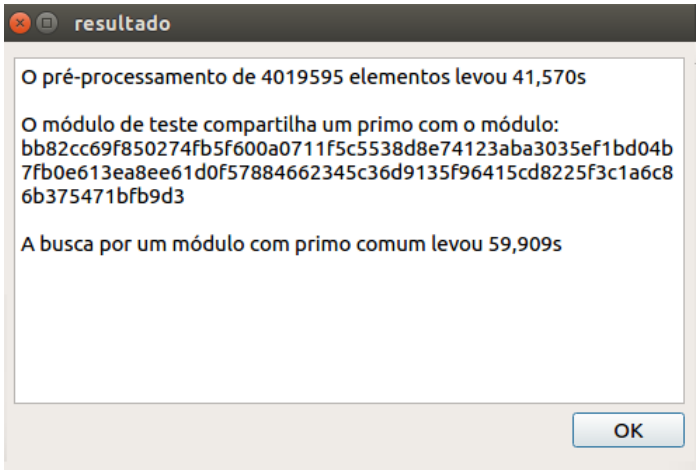

Figura 6.2: Resultado do teste de colisão entre Chave Pública e coleção de módulos RSA em hexadecimal

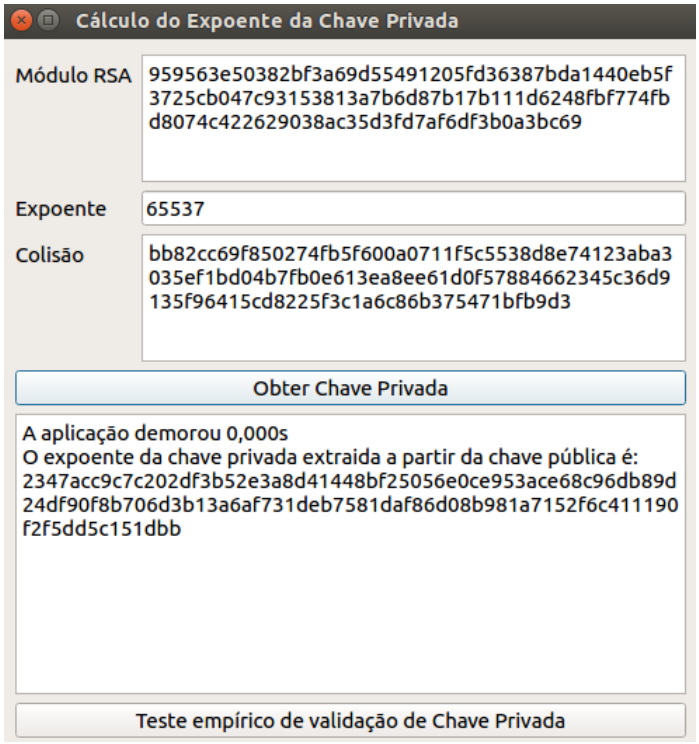

Figura 6.3: Calculo do expoente da Chave Privada

Encriptar mensagem com algoritmo RSA [12] Está opção se encontra acoplada a opção de decriptar uma mensagem com algoritmo RSA. Aqui, o usuário será capaz de observar a criptografia assimétrica em ação. A interface gráfica irá apresentar os dados referentes a chave pública do usuário e o expoente da chave privada corespondente, conforme a figura 6.4.

No quarto campo presente na interface, o usuário deve digitar a mensagem que deseja cifrar. Após o programa detectar o preenchimento de todos os campos, o botão de cifragem será habilitado. O botão de cifragem realiza a cifragem da mensagem, contida em seu referido campo, com algoritmo RSA e a chave pública disponibilizada pelo usuário. O criptograma resultante da aplicação estará disponível para visualização no campo abaixo do botão de cifragem. O usuário poderá notar que o arquivo: *cifra.tmp* foi criado, ou modificado, na pasta de instalação da ferramenta. O arquivo *cifra.tmp* contém o criptograma resultante da cifragem.

Decriptar mensagem com algoritmo RSA [12] Está opção se encontra acoplada a opção referente a cifragem de uma mensagem com algoritmo RSA. Após o pro-

[12] O código fonte da aplicação pode ser encontrado no apêndice C, página: 82.

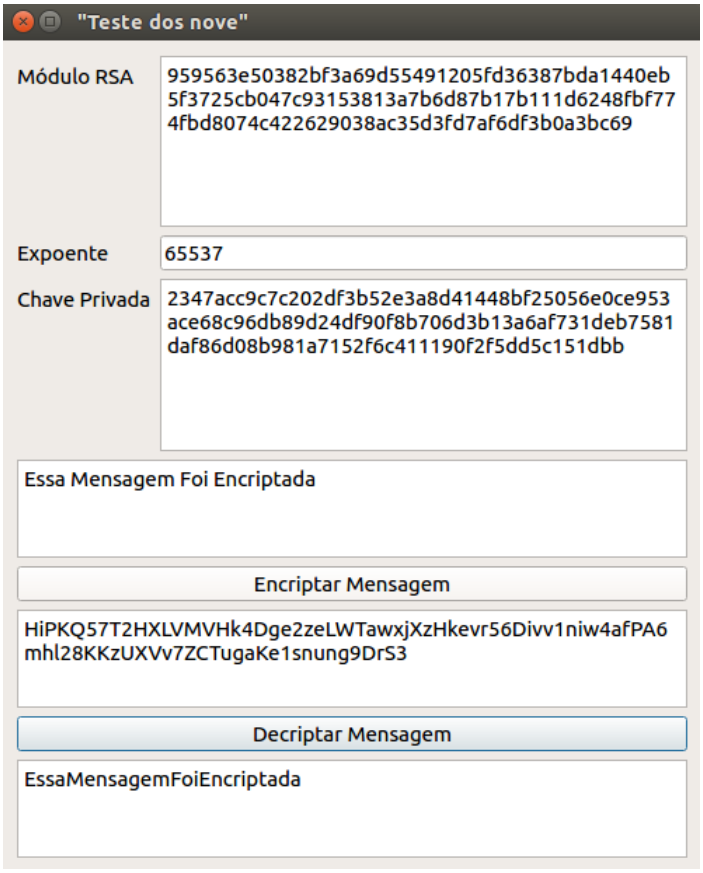

Figura 6.4: "Prova dos nove".

cesso de cifragem, mencionada no item anterior, o usuário poderá decifrar a sua mensagem, com o expoente da sua chave privada. Assim, o usuário poderá observar se o processo de teste de robustez e o processo de cálculo da sua chave privada alcançaram resultados consistentes.

Capítulo 7

ICP-Brasil

Vista a comprovação teórica exposta neste documento, de que métodos indiretos para se testar a premissa de assimetria em pares de chaves RSA, baseados na técnica proposta por Lenstra, se qualificam ao terceiro crivo da hipótese aqui tematizada, buscamos neste capítulo explorar as implicações jurídicas deste fato para a população brasileira, sujeita ao regime jurídico da ICP-BR. Como mencionado no capítulo de introdução, o uso da assinatura digital como ferramenta semiológica que substitui sob tal regime a assinatura de punho com fé pública está positivada no ordenamento jurídico brasileiro pela medida provisória (MP) 2200-2, que instituiu a ICP-BR.

No momento em que se estabelece a equivalência entre assinatura digital e assinatura de punho com fé pública, o ordenamento jurídico brasileiro impõe a responsabilidade pela comprovação de falsificação, em casos de repúdio de assinatura digital, ao suposto signatário. Com essa inversão do ônus da prova, o cidadão que alegar a falsidade de um documento apresentado em juízo contra si, contendo assinatura digital a si atribuída pela titularidade de um certificado X.509 emitido por certificadora credenciada por tal regime, fica com o ônus de provar a falsidade desta assinatura (ao invés de quem apresenta o documento, de provar a autenticidade da mesma).

Considerando o contexto da acadêmico deste trabalho, na área das ciência exatas, cabe então apresentar uma breve introdução dos termos jurídicos que serão utilizados neste capítulo. Em seguida, serão abordadas questões jurídicas de base técnica que não estão bem assentadas na MP 2200-2, e a relação destas questões com o seu desequilíbrio entre riscos e responsabilidades exposto nesse trabalho ou ampliado por seus desdobramentos. É desejo do autor que o leitor possa ter uma visão mais precisa e realista sobre a "equivalência" das assinaturas digital e de punho com fé pública, assim como

estimular o pensamento crítico quanto à real segurança alcançável na esfera digital.

7.1 Alguns termos Jurídicos

Nessa sessão, anunciaremos os termos jurídicos de relevância para o capítulo. O primeiro termo que convém descrever por referência é o já citado "ônus da prova". Seguido dos termos "medida provisória", "autarquia", "fé pública" e "prova diabólica".

7.1.1 Ônus da Prova

O ônus da prova é um termo chave para se entender a motivação deste trabalho. Assim como a validação de uma tese por intermédio de uma prova matemática, dinâmica que rege a cadeia de evolução científica, o termo em questão remete ao dever de fornecer garantias suficientes para sustentar a posição de um indivíduo [23]. Em outras palavras, toda proposição utilizada para sustentar uma posição discursiva necessita de uma prova que valide a sua corretude

7.1.2 Medida Provisória

Em casos de urgência e relevância, o Presidente da República tem o poder de instaurar uma medida provisória. Uma MP[1] [16] é um instrumento com força de Lei Federal ordinária, que produz efeitos imediatos a partir de sua publicação e que tem vigência de sessenta dias, prorrogável uma vez por igual período. O processo que efetiva a medida provisória como lei envolve a aprovação pela Câmara e pelo Senado, caso a medida seja rejeitada ela perde a sua eficácia. Após aprovação, a MP é enviada de volta para a Presidência da República para sanção.

7.1.3 Autarquia

A administração pública no Brasil divide-se em administração direta e indireta. Autarquias são entidades de administração pública indireta, criadas por meio de uma lei com finalidade específica. Autarquias são órgãos com personalidade jurídica própria

[1] http://www2.camara.leg.br/comunicacao/assessoria-de-imprensa/medida-provisoria

que desempenham funções do Estado. A receita e o patrimônio são próprios e são sujeitos a fiscalização do Estado. Como em órgãos da administração direta, os funcionários de uma autarquia devem ser aprovados por concurso público, com ressalva aos cargos comissionados em função de assessoramento, direção e chefia.

> "Autarquia – o serviço autônomo, criado por lei, com personalidade jurídica, patrimônio e receita próprios para executar atividades típicas da Administração Pública, que requeiram, para seu melhor funcionamento, gestão administrativa e financeira descentralizada."[2]

7.1.4 Fé Pública

Fé pública, segundo Silvio Rodrigues [39], refere-se a escritura pública e outros atos lavrados em cartório e servidores da justiça

> "Como goza ele de fé pública, presume-se que o conteúdo do documento seja verdadeiro, até prova em contrário." (in Direito Civil, Parte Geral, Vol.1, Saraiva, p. 268)

7.1.5 Prova Diabólica

Também conhecida por prova impossível [25] [23] ou excessivamente difícil de ser produzida[3]. Sua recorrência é evidenciada quando é necessário provar que algo não ocorreu (prova negativa). O problema da prova Diabólica gerou uma mudança na distribuição do ônus da prova no processo penal, sendo hoje adotada uma Teoria de Distribuição Dinâmica no qual o ônus da prova é atribuído a quem puder suportá-lo. É importante ressaltar que essa teoria é adotada pela doutrina e pela jurisprudência, porém não foi positivada.

7.2 Sobre a MP 2200

A MP Nº 2200-2, de 24 de agosto de 2001, instituiu a Infraestrutura de Chaves Públicas Brasileira e transformou o Instituto Nacional de Tecnologia da Informação

[2] art.5º, inciso I do Decreto Lei 200/67,`http://www.planalto.gov.br/ccivil_03/decreto-lei/Del0200.htm`(Acessado em 25/11/2017).

[3] `https://jus.com.br/artigos/21525`(Acessado em 25/11/2017)

em autarquia. Com base em suas predecessoras, as MPs Nº 2.200 de 28 de junho de 2001 e Nº 2.200-1 de 27 de julho de 2001, a MP Nº 2200-2 foi instaurada pelo presidente em exercício Fernando Henrique Cardoso. Em 11 de setembro de 2001, entrou em vigor a Emenda Constitucional Nº 32, que reformulou o processo de instauração e vigência de novas medidas provisórias, incluindo um dispositivo em seu Art 2º que diz :

> "As medidas provisórias editadas em data anterior à da publicação desta emenda continuam em vigor até que medida provisória ulterior as revogue explicitamente ou até deliberação definitiva do Congresso Nacional" [16]

Esse dispositivo assegurou a manutenção na MP 2200-2, com força de Lei Federal ordinária„ sem necessidade de reedição periódica com ligeiras alterações, artifício até então utilizado pelo Poder Executivo para MPs que não logravam aprovação no Congresso Nacional (a exemplo, neste caso, das predecessoras 2200-1 e 2200). Os artigos 1º, 5º e 6º da MP 2200-2, que desde então vigem com força de Lei Federal ordinária, instituem a hierarquia e as responsabilidades atribuídas às Entidades Certificadoras das cadeias de certificação da ICP-BR, sendo assim relevante apresentá-los aqui.

> "Art. 1º Fica instituída a Infra-estrutura de Chaves Públicas Brasileira - ICP-Brasil, para garantir a autenticidade, a integridade e a validade jurídica de documentos em forma eletrônica, das aplicações de suporte e das aplicações habilitadas que utilizem certificados digitais, bem como a realização de transações eletrônicas seguras." [17]

> "Art. 5º À AC Raiz, primeira autoridade da cadeia de certificação, executora das Políticas de Certificados e normas técnicas e operacionais aprovadas pelo Comitê Gestor da ICP-Brasil, compete emitir, expedir, distribuir, revogar e gerenciar os certificados das AC de nível imediatamente subseqüente ao seu, gerenciar a lista de certificados emitidos, revogados e vencidos, e executar atividades de fiscalização e auditoria das AC e das AR e dos prestadores de serviço habilitados na ICP, em conformidade com as diretrizes e normas técnicas estabelecidas pelo Comitê Gestor da ICP-Brasil, e exercer outras atribuições que lhe forem cometidas pela autoridade gestora de políticas." [17]

> "Art. 6º Às AC, entidades credenciadas a emitir certificados digitais vinculando pares de chaves criptográficas ao respectivo titular, compete emitir, expedir, distribuir, revogar e gerenciar os certificados, bem como colocar à disposição dos usuários listas de certificados revogados e outras informações pertinentes e manter registro de suas operações. " [17]

7.3 Desequilíbrio de riscos e responsabilidades

Tendo em vista o caráter de urgência e relevância implícitos na instauração de uma medida provisória, é notória a imposição do uso de novas tecnologias de informação e comunicação com desdobramentos jurídicos pelo ex-Presidente da República ao arrepio do Congresso Nacional. Em um período de exatos 57 dias, a MP 2200 instituiu no país a equivalência jurídica entre a assinatura de punho com fé pública e a assinatura digital sob o regime da ICP-BR. O Brasil deu um grande passo em relação ao uso da tecnologia, porém, a um custo alto em termos de insegurança jurídica. O objetivo do restante deste capítulo é expor o desequilíbrio de riscos e responsabilidades de ordem técnica causado pelo avanço forçado e precipitado do uso de novas tecnologias, pela via da norma legal imposta através de instrumento de exceção republicana [8].

7.3.1 Certificados de uso geral

O padrão X.509, adotado pela ICP-BR, suporta a edição de políticas de uso nos certificados digitais. Regimentada no Art 4º da MP-2200-2, as políticas de certificação e as regras operacionais da AC Raiz são competências do Comitê Gestor (CG) da ICP-Brasil.

> " Art. 3o A função de autoridade gestora de políticas será exercida pelo Comitê Gestor da ICP-Brasil (CG-ICP-BR), vinculado à Casa Civil da Presidência da República e composto por cinco representantes da sociedade civil, integrantes de setores interessados, designados pelo Presidente da República, e um representante de cada um dos seguintes órgãos, indicados por seus titulares:
>
> I - Ministério da Justiça;
>
> II - Ministério da Fazenda;
>
> III - Ministério do Desenvolvimento, Indústria e Comércio Exterior;
>
> IV - Ministério do Planejamento, Orçamento e Gestão;
>
> V - Ministério da Ciência e Tecnologia;
>
> VI - Casa Civil da Presidência da República; e
>
> VII - Gabinete de Segurança Institucional da Presidência da República." [17]

Uma das normas infralegais ditadas pelo CG-ICP-BR [4] estabelece como obrigatória uma política de uso geral, estratificada por nível de certificação, medida que propaga riscos de forma não linear para os titulares de certificados emitidos.

7.3.2 Risco vinculado a chave de uso geral

Uma chave de uso geral implica em que um mesmo certificado gerado com o intuito de cumprir obrigações fiscais, por exemplo, possa ser usado para validar um documento eletrônico de compra e venda de imóveis, empréstimos bancários, registro de empresas, entre outros. Não só pelo titular, o que pode ser entendido como conveniência, mas também por um eventual fraudador, o que representa risco adicional para o titular. Pois é evidente o valor atrativo que tal generalidade acumula sobre uma chave privada, gerada para uso em protocolos de criptografia assimétrica que viabilizam o esquema da assinatura digital em documentos de todo tipo sob tal regime, para um agente malicioso com intenções e condições de se apropriar sorrateira e momentaneamente da identidade do titular daquela chave, seja para locupletar-se com o uso da assinatura digital da vítima em benefício próprio, seja para criar contra ela toda sorte de problemas artificiais de natureza jurídica.

Esse risco se compõe e se amplifica pelo que está positivado no parágrafo único do Art. 6º da MP 2200-2, que diz:

> " O par de chaves criptográficas será gerado sempre pelo próprio titular e sua chave privada de assinatura será de seu exclusivo controle, uso e conhecimento." [17]

E pelo que está positivado no parágrafo primeiro do Art. 10º da mesma norma, que veremos adiante.

O primeiro atribuiu responsabilidade única ao titular de um certificado digital ICP-BR pelo controle do uso da chave privada correspondente. Esta é a única forma lógica de se interpretar tal dispositivo, tendo em vista que uma premissa semiológica de confiança – no caso, a primeira para eficácia do mecanismo criptográfico de assinatura digital – não pode ser imposta ou decretada por norma jurídica, assim como uma lei física também não pode. Enquanto o segundo, inverte o ônus da prova em caso de repúdio da assinatura digital pelo suposto signatário, quando identificado por um certificado ICP-BR que valide tal assinatura.

Primeiramente, é possível se imaginar que a apropriação indevida, em má fé ou criminosa, da chave privada de um titular de certificado, ou o controle momentâneo para uso sorrateiro da mesma, seria possível apenas mediante invasão ao computador onde esse titular opera com sua chave privada. Porém, como demonstrado pelo conteúdo desse trabalho, esta não é a única possibilidade prática. Existe a possibilidade real, no caso de chaves RSA, de dedução da chave privada a partir da chave pública correspondente, com probabilidade média de um em cada duzentos pares, considerando-se as amostras já testadas neste estudo e nos demais anteriores citados. Em outras palavras, um agente oportunista sequer precisa invadir o computador da vítima para alcançar objetivos criminosos em violação ao primeiro dispositivo. Na realidade, como ficou demonstrado, um agente oportunista sequer precisa conhecer a vítima para obter sua chave privada (desde que tenha acesso à correspondente chave pública, e o módulo desta se envolva em colisão com o de alguma outra chave RSA amostrada).

De posse da chave privada de sua vítima, o falsificador terá garantido em lei que todo e qualquer documento apresentado com a assinatura digital de sua vítima, com verificação certificada por cadeia da ICP-Brasil, terá fé pública e poderá portanto fazer prova inicial contra a vítima. Fé pública positivada pelo parágrafo primeiro do Art 10º da MP 2200-2, que diz.

> "Art. 10º Consideram-se documentos públicos ou particulares, para todos os fins legais, os documentos eletrônicos de que trata esta Medida Provisória.
>
> § 1º As declarações constantes dos documentos em forma eletrônica produzidos com a utilização de processo de certificação disponibilizado pela ICP-Brasil presumem-se verdadeiros em relação aos signatários, na forma do art. 131 da Lei no 3.071, de 1º de janeiro de 1916 - Código Civil."

Numa situação em que um agente criminoso alia competência técnica e legalista para se passar pelo titular de assinaturas digitais de um cliente da ICP-BR, este pode iniciar sorrateiramente sua prática de fraudes e ataques enquanto o cliente titular só terá pistas de sua condição de vítima quando confrontado com os efeitos e atos jurídicos praticados ou provocados por ações criminosas ou fraudulentas envolvendo falsidade ideológica. Devido à fé pública assim atribuída a documentos eletrônicos pelo Art 1º da MP 2200-2, o titular da assinatura digital nesses casos fica responsável por provar que o documento não atende à terceira premissa de confiança do esquema de assinatura digital, referente à manifestação de sua vontade no conteúdo do mesmo.

Doutro lado, a alternativa de se invocar o Código de Defesa do Consumidor, para isentar do ônus da prova de falsificação de assinatura uma alegada vítima de fraude,

com base em sua hipossuficiência ante a complexidade técnica dos objetos de tutela da MP 2200-2, ao invés de sanar esse desequilíbrio, simplesmente inverte a direção do seu efeito: Eis que tal alternativa, ou jurisprudência firmada neste sentido, pode ser invocada por legítimos signatários que queiram de má fé repudiar sua assinatura em documentos que vierem a lhes incriminar ou a lhes responsabilizar pelo cumprimento de obrigações anteriormente firmadas. Enquanto esta instabilidade jurídico-sistêmica atrai o julgador para o ativismo judicial[4], e o subjacente regime republicano, para a politização do seu Poder Judiciário.

7.4 Inversão do ônus da prova

A inversão do ônus da prova instituída pelo regime jurídico da ICP-BR mereceu o seguinte comentário do Professor Pedro A D Rezende, em palestra proferida no Primeiro Fórum de Certificação Digital organizado pelo ITI em 2003:

> "A justificativa para esta inversão, na MP 2200-2, aparenta estar desvinculada do conceito de fé pública, pois a mesma mescla interesses e atribuições públicas e privadas. É mais provável que tenha ali sido inspirada numa recomendação da UNESCO, que produziu, há cinco anos, um "modelo de lei de comércio eletrônico" [9], sob lobby da BSA (Business Software Alliance), entidade que alia os maiores players da indústria monopolista do software proprietário. " [6]

No seu relatório de pesquisa "Modelos de Confiança para Segurança em Informática", o mesmo autor inclui a seguinte análise sobre essa inversão do ônus da prova:

> "Interpretação provável dos efeitos combinados do § único do Art. 6º e do § primeiro do Art. 10º da Medida Provisória 2.200-2 de 2001, sendo, ainda, a pretendida pelo legislador à luz do Art. 1º e de opiniões publicamente circuladas por lobistas que promoveram e/ou que defendem este diploma. Seguramente a interpretação cabível sob a égide do positivismo jurídico, como em *"Internet e seus reflexos estruturais no Direito Processual"* [41]: "Dispõe o art. 116 da Lei 8112/96 (RJU) que são deveres do servidor:...VII-zelar pela conservação do material e a conservação do patrimônio público. Este item adquire especial significação no que diz respeito à utilização de hardware e software, ambas passíveis de danificação em caso de mau uso (sic.). Aqui podem ser enquadradas as condutas irregulares do funcionário que, por grave descuido, ou mesmo voluntariamente, facilita a 'infecção' de computadores e redes por códigos maliciosos, inclusive pela utilização de procedimentos e softwares em desconformidade com o marco da ICP-Brasil,

[4] `http://www.ambitojuridico.com.br/site/index.php?n_link=revista_artigos_leitura&artigo_id=16849`(Acessado em 25/11/2017)

instituída pela MP 2.200-2." Indaga-se, aqui, em que sentido a conformidade com tal marco regulatório preveniria contra infecções capazes de produzir falsificações, ou contra falsificações rastreáveis a "mau uso" na conduta do funcionário; e em que sentido a não conformidade com tal marco seria causa facilitadora de infecções capazes de causar falsificações. Sobre esta ilação, a lógica da mentalidade criminosa, que só cresce em sofisticação e especialização na esfera virtual (ver ref. [24], [40]), indica justamente o sentido contrário: haverá mais interesse em se desenvolver código malicioso, invisível e sorrateiro (ver ref. [33]), contra softwares que se conformam em dar fé pública a documentos eletrônicos (valor sob risco) do que contra softwares que não se conformam em dá-la. Sobre tal fé pública: "... [Dispõe o art. 117 da Lei 8112/96 (RJU)] que são ilícitos passíveis de responsabilização pelo servidor: III- recusar fé a documentos públicos. ... Situação ainda mais grave será a daquele servidor ou autoridade pública que negue fé quando confrontado com documento produzido por órgão público e 'assinado' com as características da ICP-Brasil, instituída pela MP-2200-2, que prescreve, em seu art. 10º:...". Cristalina, aqui, a hermenêutica produzida por um Mestre do Direito Constitucional e Advogado da União em [41]. Restou-lhe esclarecer, todavia, se tal fé pública decorre do citado artigo 10º (situação que daria fé pública também aos documentos eletrônicos privados sob o mesmo regime jurídico), ou se da titularidade da assinatura (agente de órgão público), ou se decorre de ambos." [5]

7.4.1 A iniciativa privada e o seu interesse na ICP-Brasil

O exame do potencial interesse da iniciativa privada [7] no Art 1º da MP-2200 pode ser interpretado como precaução necessária, baseada em experiências passadas. A reboque da necessidade de evolução tecnológica merecida pelo cidadão, a indústria bancária brasileira deu passos largos na tentativa de se destacar entre os seus pares no mundo, enquanto ao mesmo tempo encontrava novas maneiras de lucrar com inflação galpante e prazos folgados sob o calibre de tecnologias e práticas pré-informatização. O fato é que toda evolução abrupta na história da humanidade gerou consequências via de regra tumultuosas. No caso da indústria bancária, a proposta de revolução tecnológica gerou, e continua a perpetuar, um custo elevado com ressarcimentos para manutenção da credibilidade do sistema.

O prejuízo supracitado advém em grande parte da combinação entre o Código de Defesa do Consumidor (CDC) e o uso indevido da tecnologia por terceiros. Um indivíduo, por exemplo, que empreste o seu cartão de crédito para um terceiro realizar um saque, pode refutar a ação do terceiro acusando o banco de ter permitido a clonagem do seu cartão. Como o sistema é propriedade do banco, o CDC lhe transfere o ônus de provar que o titular do cartão realmente fez o saque. Quando incapacitado a produzir provas desta natureza, o banco fica responsável por ressarcir o cliente que

age criminosamente. A partir dessa experiência traumática da indústria bancária, fica mais compreensível sua motivação para que se institua a inversão do ônus da prova em situações tecnicamente plausíveis.

O possível conflito entre o Art 1º e o Código de Defesa do Consumidor também não foi apreciado pela MP 2200-2. O conflito é referente a já estabelecida jurisprudência contra associados, em indefensáveis posições de vulnerabilidade a fraudes, em serviços considerados indispensáveis. No documento já referenciado neste capítulo, o Professor Pedro A. D. Rezende faz a seguinte colocação:

> "Talvez não por coincidência, a Febraban[5] já havia ajuizado ação pela restrição jurisdicional do CDC, visando reverter tais posições e jurisprudência, ação que está para ser julgada no STF[6]: uma decisão favorável à Febraban eliminaria o citado risco imediato de conflito jurídico, abrindo caminho para que vija sua interpretação do parágrafo primeiro do artigo 10 da MP 2200 (inversão do ônus da prova)..." [6]

7.4.2 O problema da inversão do ônus da prova

Poucas linhas de código, ou pouca compreensão no projeto, de um software gerador de números primos pseudo-aleatórios, e um agente oportunista, podem tornar inóquo todo e qualquer grau de cautela e diligência do titular de uma chave privada para controlar o uso da mesma. Assim, a norma jurídica que atribui responsabilidade pelo uso de uma chave privada única e exclusivamente ao titular dessa chave é desfavorável e, no caso do RSA, hoje totalmente injustificada, partindo-se do pressuposto que o conhecimento necessário para a compreensão de todo o processo envolvido no protocolo de certificação digital é intangível para a grande maioria dos cidadãos que se vêem legalmente obrigados a se utilizarem desta infraestrutura para estarem em dia com suas obrigações fiscais.

O que melhor caberia, para os interesses legítimos desses cidadãos, é o direito de gerar seu par de chaves e de operar sua chave privada na plataforma computacional de

---

[5] Febraban é a principal entidade representativa do setor bancário brasileiro. Seu objetivo é representar seus associados em todas as esferas do governo (poderes Executivo, Legislativo, Judiciário e entidades representativas da sociedade). Em suma, fica evidente a preocupação da entidade com o uso da tecnologia e suas vulnerabilidades.

[6] Decisão final no STF sobre a ação em questão `http://www.stf.jus.br/portal/processo/verProcessoAndamento.asp?numero=2591&classe=ADI&origem=AP&recurso=0&tipoJulgamento=M`(Acessado em 28/11/2017)

sua escolha, sem coação ou subterfúgios para renúncia a esse direito, sob pretexto de homologações ou normas técnica ditadas por interesses de terceiros.

A inversão do ônus da prova não seria um problema tão grave se a prova diabólica fosse intangível. Ou seja, se a hipótese de uma pessoa obter a chave privada de outra, ou induzir o uso sorrateiro da mesma, só pudesse ocorrer mediante invasão do computador da vítima. Pois neste caso a vítima ainda poderia ter esperanças de o invasor ter falhado no despiste do rastro de sua invasão. Porém, como provado no conteúdo deste trabalho, um agente oportunista não precisa invadir o computador onde a vítima opera sua chave privada para ter chances reais de obtê-la, no caso de chaves RSA. Basta ter acesso legítimo a onde precisa operar sua chave pública. Dessa maneira se instaura o problema da prova diabólica para o uso desse tipo de chaves: não havendo meios para se identificar positivamente o autor do delito, o proprietário da chave privada fica com a responsabilidade e com os prejuízos causados por ações que o agente oportunista teria praticado em seu nome.

O uso geral e obrigatório de certificados de chaves RSA sob um regime normativo que atribui fé pública a documentos digitalmente assinados enquanto inverte o ônus da prova em casos de repúdio, acumula valor atrativo ao conhecimento furtivo da chave privada de seus incautos usuários, devido à utilidade desse conhecimento furtivo para a prática de crimes terceirizáveis com sofisticação e alcance difíceis de antecipar e prevenir. Privilegiar o CDC em detrimento de um tal regime, ou vice versa, nada resolve, já que isso serviria apenas para guiar a evolução dessa utilidade criminosa. Sem alteração nesse regime normativo, portanto, seu desequilíbrio sistêmico entre riscos e responsabilidades vai impondo um custo social crescente à nossa modernização digital. Será o Brasil o único país a perseguir desta forma os badalados frutos da revolução digital?

Encerramos este capítulo com mais uma citação do professor Pedro A. D. Rezende, extraída de sua apresentação no 1º Fórum de Certificação Digital organizado pelo ITI, notável pela simplicidade da figura de linguagem utilizada e pela precisão da analogia.

> "Ocorre que segurança não é orégano de pizza, que se acrescenta por cima antes de se levá-la ao forno." [6]

Capítulo 8

Considerações Finais

Procuramos o ITI com uma proposta de cooperação para analisarmos a robustez das chaves públicas homologadas no regime da ICP-Brasil, porém, não houve manifestação sobre nossa proposta, indicando desinteresse. Inicialmente, buscamos o contato por meio de Ofício (Anexo D) assinado pela chefe do Departamento de Ciência da Computação (CIC) da UnB, protocolado no ITI em 05/09/2016. O ofício propunha uma parceria onde nos prontificamos a implementar, validar e disponibilizar software estado-da-arte capaz de atender, com demanda computacional compatível com a infraestrutura do CIC ou do ITI, o acervo de certificados já emitidos sob regime da ICP-BR como amostra potencial para a execução de testes de Lenstra com suas respectivas chaves públicas RSA. O ITI seria responsável por disponibilizar o acervo de certificados de chaves públicas já emitidos sob regime da ICP-BR, para execução de testes de Lenstra, sob as condições que o órgão responsável pelo planejamento e execução de Auditorias na ICP-BR julgasse adequadas. Não obtivemos resposta do ITI nessa primeira tentativa.

Em 14 de junho de 2017, o ITI divulgou no boletim de Nº 443[1] seu plano de Dados Abertos. Neste momento, mais uma oportunidade de cooperação com a entidade reguladora foi contemplada. O plano de Dados Abertos (PDA) visa atender à política de dados abertos do Poder Executivo Federal, instituída pelo decreto número 8777 de 11 de maio de 2016, que tem por objetivo organizar e padronizar os processos de publicação de Dados Abertos de Estado. A divulgação dos dados públicos por meios eletrônicos deve primar por facilitar o reuso e permitir acesso simplificado aos usuários. Todo dado público, como regra, é público e, portanto, deve ser dado aberto.

[1] `http://iti.gov.br/dados-abertos` (último acesso realizado no dia 16/11/2017)

A leitura cuidadosa do PDA[2] não deixa claro se a disponibilização pública do acervo de Certificados Digitais estaria coberta pelo PDA. Na sessão que trata sobre a *MATRIZ DE CONJUNTO DE DADOS A SEREM PUBLICADOS* (págs.: 23-24), no setor responsável pelos repositórios da ICP Brasil, está prevista a publicação dos dados referentes à navegação (Tipos de arquivo e Downloads), e no setor responsável pela fiscalização e auditoria, está prevista a publicação dos dados referentes a Certificados Emitidos (Quantidade por ano, mês e AC). Assim, não está claro que o ITI deve ou pretende disponibilizar publicamente seu acervo de Certificados Digitais, mesmo que o PDA descreva em seu glossário o que seja considerado: "DADO PÚBLICO: qualquer dado gerado ou sob a guarda governamental que não tenha o seu acesso restrito por legislação específica."[2].

Em vista da incerteza descrita acima, houve nova tentativa de comunicação com o ITI. A partir da parceria[3] entre o ITI e a Universidade Federal de Santa Catarina (UFSC), tentamos contato com o Doutor Ricardo Custódio, supervisor do laboratório de segurança em computação da UFSC, pelo e-mail disponibilizado na página do laboratório[4], porém, nessa tentativa tampouco obtivemos qualquer resposta. O conteúdo do e-mail encontra-se na íntegra no Anexo E.

Em nova tentativa de estabelecer contato com o ITI, recorremos à intermediação da autoridade máxima da UnB. No dia 10/08/2017, foi encaminhada ao Gabinete da Reitoria (GRT) uma solicitação (Apêndice F) para que fosse encaminhado novo ofício ao Diretor-Presidente do ITI, este agora assinado pela Reitora da UnB, reiterando a proposta encaminhada ao então Diretor-Presidente em 2016. Este pedido foi motivado por recente mudança na direção do ITI, uma vez que seu atual Diretor-Presidente não é conhecido do professor orientador deste projeto, e pela possibilidade do pedido inicial ter sido desconsiderado por dirigir-se à autoridade máxima de uma autarquia federal sem ter se originado de uma autoridade máxima correspondente na UnB.

A reitoria se manifestou, em 4/10, através do chefe de Gabinete da Reitora, com instrução para que tal pedido fosse primeiramente encaminhado a instância onde se originou, para submissão a uma série de instâncias seguintes, a saber: a) Colegiado

---

[2] `http://iti.gov.br/images/repositorio/dados-abertos/Plano_de_dados_abertos_iti_bienio_2017_2018.pdf` (último acesso realizado no dia 16/11/2017)

[3] `http://iti.gov.br/noticias/indice-de-noticias/147-sistemas-de-gerenciamento-de\T1\textendashcertificados-digitais-da-icp-brasil-vao-ser-aprimorados` (último acesso realizado no dia 16/11/2017)

[4] `http://www.labsec.ufsc.br/doutores/` (último acesso realizado no dia 16/11/2017)

do Departamento de Ciência da Computação; b) Conselho do Instituto de Ciências Exatas; c) à Câmara de Ensino de Graduação, após cuja aprovação d) ao Decanato de Pesquisa e Inovação, para verificação dos cumprimentos das formalidades legais e normativas, e posteriormente, e) para Procuradoria Jurídica, para emissão de parecer. E por fim, f) de volta ao Gabinete da Reitora, para aprovação final e assinatura dos instrumentos jurídicos.

Para que a tramitação burocrática exigida pela reitoria pudesse incluir a necessária concordância da entidade proposta, em contato pessoal, intermediado por terceiros em 21 de setembro de 2017, do professor orientador desse trabalho com o atual Diretor-Presidente do ITI durante o XV Fórum de Certificação Digital organizado pela autarquia em Brasília, o mesmo lhe instruiu a reencaminhar a proposta de cooperação (Anexo D) diretamente ao seu gabinete. Iniciou-se então uma troca de emails na tentativa de finalmente obtermos resposta do ITI à presente proposta, através da qual uma comunicação neste sentido foi finalmente estabelecida. O conteúdo desses e-mails está replicado na íntegra no Apêndice G. Foram seis e-mails, pelos quais foram agendadas, em duas ocasiões, datas para uma reunião entre o professor orientador deste trabalho, Pedro A. D. Rezende, e o Diretor-Presidente do ITI, Gastão Ramos. Porém, ambas foram canceladas pelo ITI na véspera, com uma nova data para reunião proposta e aceita ante o primeiro cancelamento, e nenhuma nova data proposta pelo ITI após o segundo cancelamento.

8.1 O caso da Estônia

O uso da assinatura digital na república da Estônia foi regulamentada pelo parlamento estoniano em 8 de maio do ano 2000, através do Ato da Assinatura Digital (*Digital Signature Act*). Assim como no Brasil, a legislação aprovada pelo parlamento estoniano estabelece equivalência entre assinatura digital e assinatura de punho. A Estônia tem uma população de aproximados 1,3 milhões de habitantes, dos quais 1,275[5] milhões fazem o uso do cartão de identificação (ID-card) emitido pelo governo. Esse cartão de identificação contém um chip que utiliza criptografia assimétrica RSA para prova positiva de identificação do titular.

No dia 02 de Novembro de 2017, na *Conference on Computer and Communications Security* (CCS[6]) da *Association for Computer Machinery* (ACM), foi apresentado um

[5] `https://e-estonia.com/solutions/e-identity/` (último acesso realizado no dia 17/11/2017)

método prático de fatorização de módulos de chaves RSA, por pesquisadores da Universidade de Masaryk (Brno, República Tcheca). O artigo *The Return of Coppersmith's Attack: Practical Factorization of Widely Used RSA Moduli* [37] foi motivado pela descoberta de falha na geração de números primos para composição de chaves RSA. A falha reside em uma biblioteca desenvolvida pela fabricante de chips alemã Infineon[7], mais especificamente, na implementação do algoritmo que gera números primos nesse chip. Seguindo o princípio de divulgação responsável, os pesquisadores notificaram as autoridades estonianas responsáveis e proporcionaram tempo hábil para a implementação de medidas que buscassem solucionar a vulnerabilidade ou suas consequências, antes de divulgarem as descobertas de sua pesquisa. Desde então, o governo da Estônia revogou 750.000[8] cartões de identidade digital que estariam vulneráveis a ataques que exploram tal vulnerabilidade, e suspendeu o acesso ao acervo de chaves públicas, para prevenir abusos. No blog de notícias do programa e-Residency[9], o Diretor Kaspar Korjus publicou nota referente à potencial vulnerabilidade, apresentou o plano para solucionar o problema, e foi categórico em dizer que sempre acreditou na política de dados abertos como valor fundamental para bem guiar o programa de digitalização da Estônia.

8.2 Conclusão

O presente trabalho de graduação esclarece o problema que surge na relação entre a função semiológica de irrefutabilidade de assinaturas digitais e o desequilíbrio de riscos e responsabilidades legais imposto pela ICP-BR. Mostramos a relação lógica entre a premissa de inviolabilidade e um nível adequado de entropia no processo de geração de números primos aleatórios, para geração de chaves RSA que serão usadas em larga escala. Implementamos um algoritmo estado-da-arte para o método indireto

---

[6] `https://www.sigsac.org/ccs/CCS2017/index.html` (último acesso realizado no dia 17/11/2017)

[7] `https://www.infineon.com/cms/en/product/promopages/rsa-update/` (último acesso realizado no dia 17/11/2017)

[8] `https://arstechnica.com/information-technology/2017/10/crypto-failure-cripples\T1\textendash\millions-of-high-security-keys-750k-estonian-ids/?amp=1` (último acesso realizado no dia 17/11/2017)

[9] `https://medium.com/e-residency-blog/we-told-you-about-a-potential-security\T1\textendashvulnerability-heres-our-update-86e04119b734` (último acesso realizado no dia 17/11/2017)

que permite derivar os parâmetros de uma chave privada a partir da chave pública correspondente, mediante fatoração do módulo desse par de chaves, em caso de colisão com uma amostra significativa, detectada por varredura de MDCs. Revelamos os resultados da execução desse algoritmo em um ambiente preparado para realizar teste de robustez no acervo de chaves públicas homologadas no regime da ICP-Brasil, e desenvolvemos um programa básico para teste de robustez de chaves públicas individuais, voltado para o cidadão brasileiro comum. Totalizamos quatro tentativas frustradas de contactar o ITI, e, à luz do caso da Estônia, somos levados a avaliar possíveis motivos para o desinteresse das autoridades responsáveis pela ICP-BR na presente proposta.

De acordo com o documento que regulamenta os padrões e algoritmos criptográficos para uso sob regime da ICP-BR[10], a homologação do algoritmo de criptografia assimétrica baseado em curvas elípticas conhecido por *brainpool* , que gera chaves assimétricas sem riscos associados ao método indireto de fatoração de números inteiros, revela que, em âmbito técnico, as autoridades responsáveis pela ICP-BR estão aparentemente cientes da vulnerabilidade motivadora deste trabalho. Doutro lado, ao analisarmos a norma sobre certificados emitidos no âmbito da ICP-BR pela publicação da instrução normativa de Nº 1 de 04 de junho de 2014 (Apêndice H), observamos a deprecação de chaves para o algoritmo RSA de 1024 ou de 2048 bits, para usuário final e ACs respectivamente. E que a instrução normativa seguinte, a de Nº 3, de 10 de julho de 2014 (apêndice H), demanda a subsequente não utilização de algoritmo criptográfico RSA apenas com chave de 1024 ou 2048 bits em certificados de usuário final e de ACs respectivamente.

> "Nota (1): A função hash SHA-1 e os algoritmos criptográficos RSA 1024 bits para certificados de usuário final e RSA 2048 bits para certificados de AC NÃO DEVEM mais ser utilizados, a partir de 2012, nas emissões de certificados digitais, inclusive em suas requisições, conforme anexo II da Resolução Nº 68. Suas previsões encontram-se nos normativos da ICP-Brasil somente para preservar a compatibilidade com os certificados emitidos até o final de 2011." - Apêndice H

A partir do estudo de caso da Estônia, buscamos os revendedores de produtos Infineon no Brasil e, surpreendentemente, encontramos uma notícia referente ao ITI e produtos Infineon homologados para uso no regime da ICP-BR. Segundo o portal de notícias G1 da globo[11], o ITI afirmou que a falha detectada pelos pesquisadores da Universidade de Masaryk não foi identificada nos produtos em uso no Brasil, nem mesmo

[10] `http://www.iti.gov.br/images/repositorio/legislacao/documentos-principais/DOC-ICP-01.01---versao-3.2-PADROES-E-ALGORITMOS-CRIPTOGRAFICOS-DA-ICP-BRASIL.pdf` (último acesso realizado no dia 17/11/2017)

nos cartões homologados pela revendedora Gemalto, para fornecimento às certificadoras credenciadas na ICP-Brasil. Em nota[12], a Gemalto esclareceu que seus produtos de autenticação e de encriptação de dados não são afetados pela vulnerabilidade encontrada. Sendo assim, podemos descartar a hipótese de negligência por parte da Autoridade Certificadora Raiz (ITI), visto que esta se prontificou a responder a um portal midiático de grande penetração e influência na formação de narrativas oficiais; porém, sem nos permitir uma verificação independente de sua resposta, no âmbito da presente proposta.

Existe também a hipótese de inexistência do repositório de chaves públicas da ICP-BR; porém, considerá-la representaria conflito com a leitura técnica e objetiva do que determina o artigo 5º da MP 2200-2, que atribui responsabilidade pela "manutenção e gerência" dos certificados emitidos sob regime da ICP-BR à sua Certificadora Raiz.

Podemos também avaliar tal desinteresse como relacionado ao impacto negativo na imagem pública e/ou na atividade econômica de agentes credenciados ou credenciantes, decorríveis de eventual descoberta de falha em quantidade significativa de chaves certificadas e geradas por dispositivos homologados, ou perigosa inobservância de normas vigentes. Todavia, a partir do caso da Estônia, também podemos desconsiderar essa hipótese, haja vista a evidente importância da cooperação entre pesquisa acadêmica e autoridade responsável por infraestruturas digitais como a ICP-BR. As parcerias já firmadas no âmbito da ICP-BR, como a anteriormente citada entre o ITI e a UFSC por exemplo, poderiam, se quisessem e os priorizassem, relizar os mesmos testes oferecidos na presente proposta. Embora nada sobre tal coisa conheçamos ou encontramos divulgada.

À luz do princípio de "Fé Pública", resta-nos então concluir que o único motivo formal para o desinteresse do ITI na presente proposta sejam mesmo as "demandas de agenda do Instituto", como explicitado em e-mail enviado ao professor orientador deste trabalho de graduação (Apêndice G), cancelando *sine die* reunião para tratar da mesma. Sendo assim, embora não possamos afirmar que as autoridades brasileiras estão cientes do assunto aqui tratado e/ou de sua gravidade, podemos apontar indícios de que devem estar, mesmo estando por hora desinteressadas no tipo cooperação aqui

---

[11] `http://g1.globo.com/tecnologia/blog/seguranca-digital/post/estonia-revoga-760-mil-certificados-digitais-apos-divulgacao-de-falha.html(últimoacessorealizadonodia17/11/2017)`

[12] `https://safenet.gemalto.com/technical-support/security-updates/` (último acesso realizado no dia 17/11/2017)

proposta. Quem sabe, talvez, em ocasião futura, estimuladas pela publicidade deste trabalho, com suas inconclusas inconsistências acerca das possíveis motivações para tanto desinteresse.

Referências

[1] Random - Linux manual page. http://man7.org/linux/man-pages/man4/random.4.html. The Linux Programming Interface - uma referência e um guia detalhado para LINUX e o sistema de programação UNIX. Acessado em 09/05/2016. 31

[2] Pedro A. D. Rezende. Criptografia e segurança de dados. http://cic.unb.br/~rezende/segdados_files/CriptSeg1-2.pdf. Material do curso de Segurança de Dados, ministrado pelo Orientador desta monografia. Acessado em 06/06/2016. 2

[3] Pedro A. D. Rezende. Criptografia e segurança na informática cap. iii - iv. http://www.cic.unb.br/~rezende/segdados_files/CriptSeg3-4.pdf. Acessado em 14/09/2017. 7

[4] Pedro A. D. Rezende. Impedimento ao uso restrito de assinatura digital na icp-br. http://cic.unb.br/~rezende/trabs/impedimento.html. Acessado em 04/07/2016. 59

[5] Pedro A. D. Rezende. Modelos de confiança para segurança em informática. http://cic.unb.br/~rezende/trabs/modelos_de_confianca.pdf. Acessado em 14/09/2017. 2, 62

[6] Pedro A. D. Rezende. Privacidade e riscos num mundo de chaves públicas. http://cic.unb.br/~rezende/trabs/forumiti.htm. Acessado em 29/06/2016. 61, 63, 64

[7] Pedro A. D. Rezende. Responsabilidade e escolhas num mundo de chaves públicas. http://cic.unb.br/~rezende/trabs/ITI.htm. Acessado em 02/07/2016. 62

[8] Pedro A. D. Rezende. Totalitarismo digital. http://cic.unb.br/~rezende/trabs/ditadura.htm. Publicado em: Observatório da Imprensa em 11/7/01, Caderno internet do Jornal do Brasil em 5/7/01 (Acessado em 24/11/2017). 58

[9] Pedro A. D. Rezende. The possible laws on digital/electronic signature: On the proposed uncitral model. *SCI'2001 Proceedings*, 10:87–92, 2001. 61

[10] Boaz Barak and Shai Halevi. A model and architecture for pseudo-random generation with applications to/dev/random. In *Proceedings of the 12th ACM conference on Computer and communications security*, pages 203–212. ACM, 2005. 28

[11] Mihir Bellare, Shafi Goldwasser, and Daniele Micciancio. Pseudo-random number generation within cryptographic algorithms: The DDS case. In *Advances in Cryptology - CRYPTO'97*, pages 277–291. Springer, 1997. 37

[12] Daniel Bernstein. How to find smooth parts of integers. 2004 `http://cr.yp.to/papers.html#smoothparts.ID201a045d5bb24f43f0bd0d97fcf5355a`. Acessado em 11/04/2016. 39

[13] Daniel Bernstein. Factoring into coprimes in essentially linear time. *Journal of Algorithms*, 54(1):1–30, 2005. 38

[14] Daniel Bernstein. Fast multiplication and its applications. *Algorithmic number theory: lattices, number fields, curves and cryptography*, 44:325–384, 2008. 9, 39

[15] Dan Boneh et al. Twenty years of attacks on the RSA cryptosystem. *Notices of the AMS*, 46(2):203–213, 1999. 37

[16] BRASIL. EMENDA CONSTITUCIONAL Nº 32, DE 11 DE SETEMBRO DE 2001. Diário Oficial da União de 12/09/2001. 55, 57

[17] BRASIL. MEDIDA PROVISÓRIA No 2.200-2, DE 24 DE AGOSTO DE 2001. Diário Oficial da União de 27/08/2001. 6, 10, 57, 58, 59

[18] N.A. Carella. Note on prime gaps and very short intervals. *arXiv preprint arXiv:1008.2381*, 2010. 16

[19] PKIX Certificate and CRL Profile. Internet x.509 public key infrastructure certificate and certificate revocation list (crl) profilev. `https://tools.ietf.org/html/rfc5280`. Acessado em 11/10/2017. 5

[20] Bouke Cloostermans. Quasi-linear gcd computation and factoring rsa moduli. 2012. 9, 46

[21] Brian Conrey. The riemann hypothesis. *Notices of the AMS*, 50(3):341–353, 2003. 16

[22] Instituto Nacional de Tecnologia da Informação. Infraestrutura de chaves públicas - brasil. 2016 `http://www.iti.gov.br/icp-brasil`. Acessado em 06/06/2016. 5

[23] Fredie Didier Jr, Paula Sarno BRAGA, and Rafael Alexandria de OLIVEIRA. Curso de direito processual civil: teoria da prova, direito probatório, ações probatórias, decisão, precedente, coisa julgada e antecipação dos efeitos da tutela. *Salvador: Juspodivm*, 2, 2015. 55, 56

[24] FBI. Ic3: Internet crime complaint center annual reports. `http://www.ic3.gov/media/annualreports.aspx`. 62

[25] Cabral Fontenele. Breves considerações sobre a prova diabólica (probatio diabolica ou devil's proof). `https://jus.com.br/artigos/21525`. Acessado em 01/07/2016. 56

[26] Zvi Gutterman, Benny Pinkas, and Tzachy Reinman. Analysis of the linux random number generator. In *Security and Privacy, 2006 IEEE Symposium on*, pages 15–33. IEEE, 2006. ix, 28, 29, 32

[27] Nadia Heninger, Zakir Durumeric, Eric Wustrow, and J Alex Halderman. Mining your ps and qs: Detection of widespread weak keys in network devices. In *Presented as part of the 21st USENIX Security Symposium (USENIX Security 12)*, pages 205–220, 2012. 9, 21, 35, 42, 46

[28] Borges Junior and Eduardo Cícero Vieira. Introdução a sistemas criptográficos e o uso de geradores de sequências de números aleatórios e pseudo-aleatórios. Master's thesis, Universidade de Brasília, Brasil, 2015. 27

[29] Ronald W Langacker. *Grammar and conceptualization*, volume 14. Walter de Gruyter, 2010. 3

[30] Arjen K Lenstra, James P Hughes, Maxime Augier, Joppe Willem Bos, Thorsten Kleinjung, and Christophe Wachter. Ron was wrong, whit is right. Technical report, IACR, 2012. 14, 24, 35

[31] Arjen K Lenstra, Hendrik W Lenstra Jr, Mark S Manasse, and John M Pollard. The number field sieve. In *The development of the number field sieve*, pages 11–42. Springer, 1993. 7, 8, 46

[32] Daniel Loebenberger and Michael Nüsken. Analyzing standards for rsa integers. In *International Conference on Cryptology in Africa*, pages 260–277. Springer, 2011. 7

[33] John Marchesini, Sean W Smith, and Meiyuan Zhao. Keyjacking: Risks of the current client-side infrastructure. In *2nd Annual PKI Resarch Workshop*, 2003. 62

[34] Ilya Mironov. Factoring RSA moduli. part i. 2012 `https://windowsontheory.org/2012/05/15/979/`. Acessado em 05/06/2016. ix, 40

[35] Ilya Mironov. Factoring RSA moduli. part ii. 2012`https://windowsontheory.org/2012/05/17/factoring-rsa-moduli-part-ii/`. Acessado em 05/06/2016. 9, 14, 21, 23

[36] Louis Monier. Evaluation and comparison of two efficient probabilistic primality testing algorithms. *Theoretical Computer Science*, 12(1):97–108, 1980. 22

[37] Matus Nemec, Marek Sys, Petr Svenda, Dusan Klinec, and Vashek Matyas. The return of coppersmith's a ack: Practical factorization of widely used rsa moduli. 2017. 68

[38] Pedro A. D. Rezende. Criptografia e Segurança na Informática - Apêndice A. `http://www.cic.unb.br/~rezende/segdados_files/CriptSegA.pdf`. 15

[39] Sílvio Rodrigues. Direito civil, vol. 1. *São Paulo: Saraiva*, 1:268, 2002. 56

[40] Symantec. Internet security threat report tracks notable rise in cybercrime. www.symantec.com/about/news/release/article.jsp?prid=20060307_01. 62

[41] Luiz F T Vergueiro. Internet e seus reflexos estruturais no direito processual. Direito e Internet, Vol. II", pp. 325-354. Ed. Quartier Latin (2008). 61, 62

Apêndice A

Algoritmo responsável pelo teste de robustez

```
#include <gcdcase.h>

double now()
{
  struct timeval t;
  gettimeofday(&t, NULL);
  return (double)t.tv_sec + (double)t.tv_usec / 1000000.;
}

typedef struct vec_ {
  mpz_t *el;
  int count;
} vec_t;

// init vector v to contain count mpzs
void init_vec(vec_t *v, int count)
{
  assert(v);
  v->count = count;
  v->el = malloc(count * sizeof(mpz_t));
  assert(v->el);
  for (int i=0; i < v->count; i++)
    mpz_init(v->el[i]);
}

// free the vector v
void free_vec(vec_t *v)
{
  assert(v);
  for (int i=0; i < v->count; i++)
    mpz_clear(v->el[i]);
  free(v->el);
}
```

```
// initializes vec_t *v and fills it with contents of named binary
    format file
void input_bin_array(vec_t *v, char *filename)
{
  fprintf(stderr, "lendo %s...\n", filename);
  FILE *in = fopen(filename, "rb");
  assert(in);
  int count;
  int ret = fread(&count, sizeof(count), 1, in);
  assert(ret == 1);
  assert(count >= 0);
  init_vec(v, count);

  //contagem de bytes lidos
  size_t bytes = 0;
  size_t sum = 0;
  for (int i=0; i < count; i++){
    bytes = mpz_inp_raw(v->el[i], in);
    assert(bytes > 0);
    sum += bytes;
  }
  fclose(in);
  fprintf(stderr, "%d elementos, %zu bytes \n", v->count, sum);

}

//::::::::::::::::::: Teste de Robustez ::::::::::::::::::://

int gcd_case4(char *f_in, char *module_in, char *exponent_in){
    double start = now();
    mpz_t t;
    mpz_init(t);
//::::::::::::::::::: tratamento de input ::::::::::::::://
    FILE *data_base = fopen(f_in, "r");
    assert(data_base);
    FILE *out = fopen("input.tmp", "wb");
    assert(out);
    FILE *report = fopen("report.txt", "wr");
    FILE *colisao_input = fopen("colisao.input", "wr");

    int count=0;
    fwrite(&count, sizeof(count), 1, out);

    for (;;) {
        int res = gmp_fscanf(data_base, "%Zx", t);
        if (res == EOF) break;
        if (res != 1) {
            fprintf(report,"Erro na linha %d do arquivo de input\n"
                ,count);
        }
        mpz_out_raw(out,t);
        count++;
    }
```

```
    fclose(data\_base);
    rewind(out);
    fwrite(&count, sizeof(count), 1, out);
    fclose(out);

    fprintf(report,"O pré-processamento de %d elementos levou \%0.3
        fs\n\n", count, now()-start);

//::::::::::::::::::: inicia vetor de inputs ::::::::::::::::://

    vec_t moduli;
    input_bin_array(&moduli,"input.tmp");

//::::::::::::::::::: Colisao ::::::::::::::::::::::::::::::://

    vec_t rsa_moduli;
    init_vec(&rsa_moduli, 2);

    gmp_sscanf(module_in, "%Zx", rsa_moduli.el[0]);              //m
        ódulo rsa
    gmp_sscanf(exponent_in, "%Zx", rsa_moduli.el[1]);
                //valor do expoente

    vec_t test;
    init_vec(&test, 2);

    int i = 0;
    int flag_nf = 0;
    for(; i < count; i++){
        mpz_set(test.el[0],rsa_moduli.el[0]);
       mpz_gcd(test.el[0], test.el[0], moduli.el[i]);
        if(mpz_cmp_ui(test.el[0],1) && mpz_cmp(rsa_moduli.el[0],
            moduli.el[i])){
            gmp_fprintf(report,"O módulo de teste compartilha um
                primo com o módulo: %Zx\n", moduli.el[i] );
            gmp_fprintf(report, "\n A busca por um módulo com primo
                 comum levou %0.3fs\n", now()\-start);

            //arquivo para input do módulo de quebra
            gmp_fprintf(colisao_input,"%Zx\n",rsa_moduli.el[0]);
            gmp_fprintf(colisao_input,"%Zx\n",rsa_moduli.el[1]);
            gmp_fprintf(colisao_input,"%Zx\n",moduli.el[i]);
            i = count;
            flag_nf = 1;
        }
    }

    if(flag_nf == 0){
        fprintf(report, "%s\n", "O módulo de teste nao compartilha
            um primo com nenhum módulo da malha de testes" );
        fclose(report);
        fclose(colisao_input);
        return 0;
    }
```

```
    fclose(report);
    fclose(colisao_input);
    return 1;
}
```

Apêndice B

Algoritmo responsável pelo calculo da Chave Privada

```
//::::::::::::::::::: Cálculo da Chave privada ::::::::::::://

int gcd_case1(char *mod_in, char *exp_in, char *colisao_in){
    double start = now();

//::::::::::::::::::: Tratamento de input ::::::::::::::::://
    vec_t colission_moduli;
    init_vec(&colission_moduli, 3);

    gmp_sscanf(mod_in, "%Zx", colission_moduli.el[0]);          //m
       ódulo rsa
    gmp_sscanf(exp_in, "%Zx", colission_moduli.el[1]);          //
       valor do expoente
    gmp_sscanf(colisao_in, "%Zx", colission_moduli.el[2]);
               //módulo de colisao

//::::::::::::::::::: Euclides Estendido ::::::::::::::::::://

    vec_t test;
    init_vec(&test, 2);

    vec_t s,t,g,a,b;
    // g = 1
    // s = Chave Privada
    // t = coeficiente
    // a = expoente da chave pública
    // b = (p-1)(q-1)

    init_vec(&s,1);
    init_vec(&t,1);
    init_vec(&g,1);
    init_vec(&a,1);
    init_vec(&b,1);
```

```
    mpz_set(a.el[0],colission_moduli.el[1]); //setting expoente da
        chave pública
    mpz_gcd(test.el[0], colission_moduli.el[0], colission_moduli.el
        [2]);     //get p

    mpz_divexact(test.el[1],colission_moduli.el[0],test.el[0]);
                    //get q
    mpz_sub_ui(test.el[1],test.el[1],1);
    mpz_sub_ui(test.el[0],test.el[0],1);
    mpz_mul(b.el[0],test.el[0],test.el[1]);      //setting (p-1)(q
        -1)

    mpz_set_ui(g.el[0],2);
    mpz_gcdext (g.el[0],s.el[0],t.el[0],a.el[0],b.el[0]);

    gmp_printf("O expoente da chave privada extraida a partir da
        chave pública é: %Zx\n\n", s.el[0], "(o módulo é o mesmo)" )
        ;
    FILE *private_tmp = fopen("private.tmp","wr");

    gmp_fprintf(private_tmp,"A aplicacao demorou %0.3fs\n",now()-
        start);
    gmp_fprintf(private_tmp,"O expoente da chave privada extraida a
         partir da chave pública é:\n%Zx",s.el[0]);
    fclose(private_tmp);

    free_vec(&b);
    free_vec(&a);
    free_vec(&g);
    free_vec(&t);
    free_vec(&s);
    free_vec(&test);
    free_vec(&colission_moduli);
    return 1;
}
```

Apêndice C

Algoritmo responsável por encriptar uma mensagem

```
//::::::::::::::::::: Encriptar Mensagem ::::::::::::://

int gcd_case2(char *msg_in, char *mod_in, char *exp_in, int opcode)
   {
    vec_t rsa_moduli;
    init_vec(&rsa_moduli, 2);

    gmp_sscanf(mod_in, "%Zx", rsa_moduli.el[0]);                  //m
       ódulo rsa
    gmp_sscanf(exp_in, "%Zx", rsa_moduli.el[1]);                  //
       valor do expoente

    mpz_t u;
    mpz_init(u);
    char buffer[1000];

        if(opcode == 1){
                strcpy(buffer,msg_in);
        }
        else{
                FILE *msg_file = fopen(msg_in, "r");
                assert(msg_file);

        gmp_fscanf(msg_file, "%[0-9a-zA-Z ]s", buffer);
                fclose(msg_file);
        }
    mpz_set_str(u,buffer,62);

//::::::::::::::::::: Criptografia Assimétrica :::::::::://
    vec_t criptograma;
    init_vec(&criptograma,1);
    mpz_powm_sec(criptograma.el[0],u,rsa_moduli.el[1],rsa_moduli.el
       [0]);

    FILE *msg_tmp = fopen("cifra.tmp","wr");
```

```
    mpz_out_str(msg_tmp,62,criptograma.el[0]);

    fclose(msg_tmp);
    free_vec(&rsa_moduli);
    free_vec(&criptograma);
    return 1;
}

//::::::::::::::::: Decriptar Mensagem ::::::::::::://

int gcd_case3(char *msg_in, char *mod_in, char *private_in){
    mpz_t u;
    mpz_init(u);
    char buffer[1000];

    FILE *msg_file = fopen(msg_in, "r");
    assert(msg_file);

    gmp_fscanf(msg_file, "%[0-9a-zA-Z ]s", buffer);
    fclose(msg_file);

    mpz_set_str(u,buffer,62);

    //::::::: Tratamento do arquivo de entrada :::::::://
    vec_t keys;
    init_vec(&keys, 2);

    gmp_sscanf(mod_in, "%Zx", keys.el[0]);              //módulo da
        chave pública
    gmp_sscanf(private_in, "%Zx", keys.el[1]);          //módulo da
        chave privada

    vec_t decriptado;
    init_vec(&decriptado,1);

    mpz_powm_sec(decriptado.el[0],u,keys.el[1],keys.el[0]);

    FILE *msg_tmp = fopen("decriptado.tmp","wr");
    mpz_out_str(msg_tmp,62,decriptado.el[0]);

    fclose(msg_tmp);
    mpz_clear(u);
    free_vec(&decriptado);
    free_vec(&keys);
    return 1;
}
```

Apêndice D

Ofício encaminhado para o ITI

Ofício CIC-UnB 01/2016

Brasília, 2 de Setembro de 2016

De: Chefe do Departamento de Ciência da Computação da Universidade de Brasília (CIC-UnB)
Prof. André Costa Drummond
Para: Diretor-Presidente do Instituto Nacional de Tecnologia da Informação (ITI)
Sr. Renato Martini
Assunto: Proposta colaborativa para aprimoramento de processo homologatório
Ref.: Infraestrutura de Chaves Públicas Brasileira (ICP-BR)

Recebido Protocolo ITI
Em 05/09/16
Às 12:09

Senhor Diretor-Presidente,

Considerando que as normas e procedimentos para Avaliação e Conformidade de equipamentos e de sistemas homologados para operar no regime da ICP-BR estão, conforme descrito em http://www.iti.gov.br/servicos/homologacoes, em processo contínuo de evolução adaptativa, e a justificativa insculpida na Instrução Normativa Nº 02/2014, de julho de 2014, de que "*a evolução tecnológica tem proporcionado o surgimento de novos dispositivos criptográficos não abrangidos pelos atuais Manuais de Conduta Técnica*";

Vimos, através do presente ofício, expor, propor e justificar o seguinte:

Tendo em vista que:

- Dentre as novidades criptográficas proporcionadas pela evolução tecnológica, surgem também implementações de técnicas para detecção – e/ou sua correspondente dual, para exploração clandestina – de vulnerabilidades até então desconhecidas ou indetectáveis em infraestruturas que tenham objetivo, porte e escopo equivalentes aos da ICP-BR;
- Dentre essas técnicas, tornou-se pública em 2012 uma metodologia para teste massivo de robustez de chaves RSA, descrita por uma equipe liderada pelo cientista Arjen Lenstra[1], em paralelo à sua correspondente dual – que representa risco sistêmico –, a qual explora potenciais falhas conceituais, de implementação, de integração ou para degradação ofuscada, responsáveis por insuficiência de entropia no funcionamento de geradores pseudorandômicos (PRNG), usados na geração de pares de chaves assimétricas em regime de produção;
- Dentre os procedimentos vigentes no regime normativo da ICP-BR para Avaliação e Conformidade de dispositivos e sistemas, o objeto da análise homologatória são artefatos individuais, modelos ou exemplares de

1 Lenstra, Arjen et. al: "*Ron was wrong, Whit is right*" Em https://eprint.iacr.org/2012/064.pdf

hardware ou de software; fato que limita esses procedimentos pontuais, quanto a poderem detectar insuficiência entrópica na geração de chaves, quando esta insuficiência tem origem em efeitos ou subterfúgios ainda desconhecidos ou desconhecíveis do testador, do autor do artefato ou da metodologia homologatória. Enquanto, por outro lado, esse tipo de insuficiência pode ser empiricamente detectada pela metodologia proposta por Lenstra – já testada com preocupante êxito no contexto global de alguns protocolos[1] – com acurácia proporcional ao tamanho da amostra de chaves públicas submetida ao teste;

- Dentre as possíveis explorações desse tipo de insuficiência, em infraestruturas da natureza e do escopo da ICP-Brasil, a mais grave consiste em se buscar, a partir de um alvo e/ou numa amostra suficiente, duas chaves públicas cujos módulos tenham divisor comum maior que 1, o qual permitiria facilmente calcular as respectivas chaves privadas. Qualquer sucesso clandestino nessa busca permite ataques que, no âmbito da ICP-BR, quebram a premissa técnico-jurídica insculpida no § único do Art. 6º da MP 2200-2, cujas consequências danosas – via falsificabilidade indetectável de assinaturas digitais correspondentes – são amplificadas pela incidência do instituto da fé pública aduzida pelo § 2º, Art. 10º do mesmo diploma legal
- Em contrapartida, se a mesma técnica for empregada como método homologatório adicional nesse mesmo âmbito, isso permitiria ao gestor técnico-normativo da ICP-BR não só antecipar-se a ataques desse tipo, neutralizando-os com Auditoria seguida do recurso jurídico da revogação, por razões técnicas, de eventuais certificados com chaves detectadas como quebráveis por meio desse tipo de teste – por descoberta de colisão entre primos semeados por PRNGs –, como também o permitiria direcionar melhor outros procedimentos de Avaliação e Conformidade, aprofundando-os sobre os artefatos vinculados aos piores indícios de insuficiência entróprica, sinalizada por maiores taxas de colisão entre as chaves que estes geram.

Temos a propor:

- Uma parceria entre a UnB – através de seu Departamento CIC – e o ITI, com vistas a integrar, ao planejamento das atividades de Auditoria sob responsabilidade do ITI, o projeto de pesquisa em curso sobre o tema no CIC, objetivando viabilizar, para a ICP-Brasil, o efetivo valor do método aqui nomeado "teste de Lenstra" como procedimento complementar de Auditoria, uma vez que o reputamos essencial para uma boa gerência do perfil de riscos associado ao uso institucional da criptografia assimétrica no cenário global contemporâneo.
- Que essa parceria seja instituída da seguinte forma:

  1. *Por parte do citado projeto de pesquisa em andamento no CIC*:
     Com a implementação, validação e disponibilização de software estado-da-arte capaz de atender, com demanda computacional compatível com a infraestrutura deste, o acervo de certificados já emitidos sob o regime da ICP-BR como amostra potencial e ideal para a execução de testes de Lenstra com suas respectivas chaves públicas.

  2. *Por parte do ITI*:
     Com a disponibilização do acervo de certificados de chave pública já emitidos sob o regime da MP 2200-2, para execução de testes de Lenstra, sob as condições que o órgão responsável pelo planejamento e execução de Auditorias na ICP-BR julgar adequadas.

Sob as seguintes justificativas:

- Durante a primeira fase da referida pesquisa, orientada pelo prof. Pedro A. D. Rezende com a participação do aluno João Henrique Sousa em trabalho de graduação concluído, na qual foram replicadas as condições gerais

para testes em escala piloto, com implementação *default*, processamento em lotes, e amostra de certificados disponíveis via https coletada pela ONG *Electronic Frontier Foundation* já armazenada em repositório público[2], constatou-se que a parte dos certificados emitidos no âmbito da ICP-BR perfazia apenas 1 milésimo da amostra escolhida (1153 em 1 milhão); insuficiente, portanto, para resultados estatísticos específicos em face da referência, que era a taxa de colisões na amostra pioneira coletada e testada por Lenstra[1];

- Para a segunda fase da referida pesquisa, sob o mesmo orientador e com a participação do aluno Ivan Sena em trabalho de graduação por concluir, as condições são adequadas e conducentes à realização da parceria aqui proposta: A plataforma para realização de testes nesta fase, já validada pelo aluno que a implementou, se baseia em biblioteca livre que codifica o estado-da-arte em eficiência para cálculo de divisor comum em larga escala, de complexidade subquadrática e de autoria da equipe que até hoje lidera em abrangência a aplicação do "teste de Lehman" (Haldemann et. al.)[3], com 24 milhões de chaves coletadas da web testadas em poucas horas, em hardware com capacidade comparável ao disponível para esta pesquisa;

- Assim, a referida pesquisa já está adequadamente instrumentada para atender sua parte nessa proposta, considerando-se o tamanho do acervo de certificados da ICP-BR, estimado pelo presidente da Associação de Autoridades de Registro (AARB) em cerca de 8 milhões, durante debate em que participavam como convidados o Diretor-Presidente do ITI e o orientador dessa pesquisa[4], em evento realizado pelo Comitê Gestor da Internet do Brasil. Bastando portanto, para sua concretização, que o ITI disponibilize tal acervo sob condições que julgar adequadas, uma vez que a estimada grande maioria dos certificados neste acervo foram emitidos para fins de assinatura digital, e por isso não estariam normalmente disponíveis na web

- Tendo em vista que foram infrutíferas as tentativas prévias do orientador dessa pesquisa em estabelecer tal parceria diretamente com a AARB, considerando-se que uma leitura coerente do disposto no Art. 5º da MP 2200-2 – sobre a responsabilidade por "gerenciar a lista de certificados emitidos" (adicionalmente à dos emitidos pela própria AC Raiz) – indica o ITI como parceiro institucional e natural do objeto deste Ofício, e considerando-se, alternativamente, que a Lei Nº 12.527 de 2011 (LAI) poderia ser acionada para viabilizar proposta equivalente menos colaborativa; ou pior, por ilegítimos interessados em possíveis insuficiências entrópicas neste acervo, dispostos a acionar inclusive expedientes extralegais;

  Em resumo, propugnando pela urgência e caráter publicamente benéfico desta proposta.

No aguardo da vossa manifestação, subscrevo-me.

Respeitosamente,

André Costa Drummond
Chefe do Departamento de Ciência da Computação da UnB

Prof. Dr. André Drummond
Matrícula: 1050711
Chefe do Dep. de Ciência da Computação - CIC
Universidade de Brasília - UnB

2 The EFF SSL Observatory Em https://www.eff.org/observatory.
3 Haldemann et. al. "Minding your ps and qs:..." Em https://factorable.net/weakkeys12.extended.pdf
4 Em https://www.youtube.com/watch?v=L-TWnc2zvBc 38m:05s

Apêndice E

E-mail enviado ao Doutor Ricardo Custódio no dia 05 de Julho

De: Ivan Sena <ivansena35@gmail.com>
Data: qua, 5 de jul de 2017 às 10:48
Assunto: Aprimoramento do gerenciamento dos certificados digitais
Para: <custodio@inf.ufsc.br>

Boa tarde Doutor Ricardo,

sou aluno de bacharelado em Ciência da Computação na UnB e, motivado pelo artigo: "Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices"de J. Alex Halderman, da University of Michigan, comecei um estudo sobre certificados digitais, a ICP-Brasil e o ITI. Orientada pelo Professor Pedro Rezende, estou desenvolvendo uma tese referente a obtenção de chaves privadas a partir de chaves públicas "fracas". Em uma visita recente ao sítio do ITI, encontrei uma notícia referente ao aprimoramento do gerenciamento dos certificados digitais por intermédio de uma parceria com a UFSC e o Laboratório de Segurança em Computação. Também encontrei uma notícia referente ao PDA (plano de dados abertos) que será promovido pelo ITI para atender a política de dados abertos do Pedor Executivo Federal. Assim, gostaria de pedir o apoio do senhor e do seu laboratório para finalizar a minha tese de graduação.

O departamento de Ciência da Computação da UnB encaminhou um ofício ao ITI, em 5/9/2016, referente ao acesso ao repositório de todos os certs já emitidos pela ICP-Brasil, nos termos do art. 5 da MP 2200-2, para fins de teste quanto à robustez das respectivas chaves, a ser executado em ambiente controlado e supervisionado por autoridade da ICP-BR designada. O mesmo ofício ainda não foi respondido. Acredito que após as notícias publicadas no portal do ITI, sobre parceria com a UFSC, e de uma resposta positiva ao meu pedido de apoio ao senhor, eu possa finalizar a minha tese de graduação com um resultado construtivo para todas as instituições encolvidas. Obrigado pela atenção,

Ivan Menezes Sena

Apêndice F

Despacho encaminhado para assessoria da GRT

Centro de custo: Departamento de Ciência da Computação (CIC)

Para: Gabinete da Reitora,

Encaminho aqui o pedido mencionado em telefonema do signatário, prof. **Pedro Antonio Dourado de Rezende**, para a Sra. **Dioney Magalhães Brito** em 10/08/2017, solicitando à assessoria do GRT que, em havendo concordância, prepare e encaminhe ofício assinado pela Reitora da UnB ao Diretor-Presidente do Instituto de Tecnologia da Informação (ITI), Sr. Gastão Ramos, reiterando, em nome da UnB, a proposta encaminhada ao então Diretor-Presidente em 2016, cuja cópia acompanha este processo (documento **1504054**).

Trata-se de uma proposta de colaboração em pesquisa acadêmica orientada pelo signatário, destinada a produzir resultados técnicos no âmbito da Infraestrutura de Chaves Públicas Brasileira (ICP-BR), sob o escopo normativo da Medida Provisória 2200 (que instituiu o ITI como autarquia gestora da ICP-BR). Tal proposta, que foi assinada pelo chefe de Departamento (CIC-UnB), encaminhada pelo professor interessado e protocolada no ITI em 5 de setembro de 2016, atá agora não foi respondida.

Decidimos encaminhar inicialmente a proposta assinada pela chefia do CIC pelo fato do então Diretor-Presidente do ITI, e o professor do CIC interessado na referida pesquisa, serem conhecidos, tendo inclusive trabalhado juntos no ITI em 2005 quando o mesmo professor foi cedido pelo então Reitor da UnB para assessorar a presidência daquela autarquia, a pedido desta, por um ano.

Houve inclusive uma segunda tentativa de iniciar tal colaboração, em 5 de julho deste ano, esta por via menos formal, através de contato direto entre um dos alunos orientandos deste professor e um professor da Universidade Federal de Santa Catarina que dirige laboratório de pesquisa já envolvido em projeto colaborativo com o ITI (destinado melhorar a trasparência dos dados geridos pela autarquia), que tampouco foi respondido até o momento (documento **1504075**).

Este pedido ao GRT se justifica pela recente mudança na direção do ITI, uma vez que seu atual diretor-presidente não é conhecido deste professor do CIC, e pela possibilidade do pedido inicial ter sido desconsiderado por dirigir-se à autoridade máxima de uma autarquia federal sem ter se originado de uma autoridade máxima correspondente na UnB.

Esperando poder contar com seu valioso préstimo, subscrevo-me.

Atenciosamente,

Pedro Antonio Dourado de Rezende

Departamento de Ciencia da Computação

Apêndice G

A reunião que nunca aconteceu

Em 21-09-2017 15:47, Pedro A.D.Rezende escreveu:
Ao chefe de gabinete do diretor-presidente do ITI,

Prezado senhor

Fomos apresentados no encontro no 15 Certforum, quando tratamos do encaminhamento da proposta em anexo. Da parte do Diretor-Presidente do ITI, recebi na ocaisão instrução para encaminha-la ao gabinete, pelo que segue anexo, com vistas a, inteirados do seu conteúdo, possamos marcar uma reunião no ITI para tratar do assunto. Assim, fico no aguardo de sua proposta de agendamento da referida reunião.
Atenciosamente,

---

prof. Pedro Antonio Dourado de Rezende
Computacao - Universidade de Brasilia
tcp: Libertatis quid superest digitis serva
http://www.cic.unb.br/docentes/pedro/sd.php

---//---

Em 26-09-2017 11:27, iti.gabinete escreveu:

Prezado Senhor,

Em atendimento a sua solicitação de audiência, incumbiu-me o Diretor-Presidente do Instituto Nacional de Tecnologia da Informação, Gastão Ramos, de informá-lo que o receberá nesta reunião, conforme a seguir:

Data: 29/09/2017

Horário: 10h00

Local: Sala de Reunião do Gabinete

Queira, por gentileza, confirmar esta reunião, bem como discriminar nominalmente demais participantes que venham a compor o referido evento pelo e-mail: iti.gabinete@iti.gov.br
Atenciosamente,

ITI - Instituto Nacional de Tecnologia da Informação

Gabinete da Presidência/ITI
Casa Civil
Presidência da República
+55 61 3424-3875
SCN Quadra 2 bloco E
70712-905 | Brasília/DF
Descrição: Facebook Descrição: Instagram Descrição: Twitter Descrição: YouTube Descrição: Blog

————————————————————//————————————————————

Em 26-09-2017 11:58, Pedro A.D.Rezende escreveu:
Sim, confirmado sexta-feira 29/09 10h. Obrigado

————————————————————//————————————————————

Em 28-09-2017 17:43, iti.gabinete escreveu:

Prezado Senhor,

Agradecendo sua especial atenção ao nosso convite, informo que por incompatibilidade de agenda, a presente reunião está cancelada. Segue sugestão para nova data:

Data: 03/10/2017

Horário: 10h00

Local: Sala de Reunião do Gabinete

Queira, por gentileza, confirmar se está de acordo com a data sugerida. Atenciosamente,
ITI - Instituto Nacional de Tecnologia da Informação

Gabinete da Presidência/ITI
Casa Civil
Presidência da República

+55 61 3424-3875
SCN Quadra 2 bloco E
70712-905 | Brasília/DF
Descrição: Facebook Descrição: Instagram Descrição: Twitter Descrição: YouTube Descrição: Blog

————————————————————//————————————————————

Em 30-09-2017 13:36, Pedro A.D.Rezende escreveu:
Sugestão aceita, confirmo interesse, Agradecido

————————————————————//————————————————————

Assunto: Re: (CANCELAMENTO DE REUNIÃO) contato referente a proposta de colaboração na UnB
Data: Mon, 2 Oct 2017 11:00:36 -0300
De: Pedro A.D.Rezende <prezende@unb.br>
Responder a: prezende@unb.br
Para: ITI GABINETE CORPORATIVO <iti.gabinete@iti.gov.br>
98124XXXX [Anonimizado nesta réplica]

Em 02-10-2017 09:30, ITI GABINETE CORPORATIVO escreveu:

Prezado Professor Pedro Rezende,

Informamos que a agenda está cancelada devido a demandas de agenda do Instituto. Entraremos em contato para marcarmos uma nova data. Pedimos que o senhor nos envie o seu número de celular para contato direto. Respeitosamente,

Lidiane Antunes
ITI - Instituto Nacional de
Tecnologia da Informação

Gabinete da Presidência/ITI
Casa Civil
Presidência da República
+55 61 3424-3875
SCN Quadra 2 bloco E
70712-905 | Brasília/DF
Descrição: Facebook Descrição: Instagram Descrição: Twitter Descrição: YouTube Descrição: Blog

Apêndice H

Regulamentação da Criptografia de Curvas Elípticas Brainpool para geração de Chaves Assimétricas no âmbito da ICP-BRASIL

INSTITUTO NACIONAL DE TECNOLOGIA DA INFORMAÇÃO

INSTRUÇÃO NORMATIVA Nº 01, DE 04 DE JUNHO DE 2014.

REGULAMENTA A CRIPTOGRAFIA DE CURVAS ELÍPTICAS BRAINPOOL PARA GERAÇÃO DE CHAVES ASSIMÉTRICAS NO ÂMBITO DA ICP-BRASIL (DOC-ICP-01.01).

O DIRETOR-PRESIDENTE DO INSTITUTO NACIONAL DE TECNOLOGIA DA INFORMAÇÃO, no uso das atribuições que lhe foram conferidas pelo inciso I, do art. 1º, do anexo I, do Decreto nº 4.689, de 7 de maio de 2003, e pelo art. 1º da Resolução nº 33 do Comitê Gestor da ICP-Brasil, de 21 de outubro de 2004;

CONSIDERANDO a necessidade de robustecimento dos padrões de algoritmos criptográficos baseados em criptografia de curvas elípticas regulamentados no âmbito da ICP-Brasil;

R E S O L V E :

Art. 1º Alterar a Tabela "Geração de Chaves Assimétricas de AC" do item 2 do DOC-ICP-01.01, versão 2.3, que passa a vigorar com a seguinte redação:

Geração de Chaves Assimétricas de AC	
Normativo ICP-Brasil	DOC-ICP-01 - item 6.1.1.3 DOC-ICP-04 - item 6.1.1.3 DOC-ICP-01 - item 6.1.5 DOC-ICP-05 - item 6.1.5
Algoritmo	RSA, ECC-Brainpool (conforme RFC 5639)
Tamanho de chave	RSA 2048, RSA 4096, brainpoolP512r1

Art. 2º Alterar a Tabela "Geração de Chaves Assimétricas de Usuário Final" do item 2 do DOC-ICP-01.01, versão 2.3, que passa a vigorar com a seguinte redação:

Geração de Chaves Assimétricas de Usuário Final	
Normativo ICP-Brasil	DOC-ICP-04 - item 6.1.5.2
Algoritmo	RSA, ECC-Brainpool (conforme RFC 5639)
Tamanho de chave A1, A2, A3, S1, S2, S3, T3	RSA 1024, RSA 2048, brainpoolP256r1
Tamanho da chave A4, S4, T4	RSA 2048, RSA 4096, brainpoolP512r1

Art. 3° Fica aprovada a versão 2.4 do documento PADRÕES E ALGORITMOS CRIPTOGRÁFICOS DA ICP-BRASIL (DOC-ICP-01.01).

§ 1º Todas as demais cláusulas do DOC-ICP-01.01, na sua versão 2.3, em sua ordem originária, integram a presente versão 2.4 e mantêm-se válidas.

§ 2º O documento referido no caput encontra-se disponibilizado, em sua totalidade, no sítio http://www.iti.gov.br.

Art. 4º Esta Instrução Normativa entra em vigor na data de sua publicação.

RENATO DA SILVEIRA MARTINI

INSTITUTO NACIONAL DE TECNOLOGIA DA INFORMAÇÃO

INSTRUÇÃO NORMATIVA Nº 03, DE 10 DE JULHO DE 2014.

ESCLARECE A MANUTENÇÃO DE SHA1 E O TAMANHO DE CHAVES RSA PARA PRESERVAR COMPATIBILIDADE COM CERTIFICADOS EMITIDOS ANTES DE 2012.

O DIRETOR-PRESIDENTE DO INSTITUTO NACIONAL DE TECNOLOGIA DA INFORMAÇÃO, no uso das atribuições que lhe foram conferidas pelo inciso I, do art. 1º, do anexo I, do Decreto nº 4.689, de 7 de maio de 2003, e pelo art. 1º da Resolução nº 33 do Comitê Gestor da ICP-Brasil, de 21 de outubro de 2004;

CONSIDERANDO a necessidade de esclarecer motivo para a manutenção dos padrões de algoritmos criptográficos para preservar a compatibilidade com os certificados até o ano de 2011;

R E S O L V E :

Art. 1º Acrescenta-se a NOTA (1) ao item 2, do DOC-ICP-01.01, versão 2.4, com a seguinte redação:

> ***Nota (1)****: A função hash SHA-1 e os algoritmos criptográficos RSA 1024 bits para certificados de usuário final e RSA 2048 bits para certificados de AC NÃO DEVEM mais ser utilizados, a partir de 2012, nas emissões de certificados digitais, inclusive em suas requisições, conforme anexo II da Resolução n° 68. Suas previsões encontram-se nos normativos da ICP-Brasil somente para preservar a compatibilidade com os certificados emitidos até o final de 2011*.

Art. 2° Fica aprovada a versão 2.5 do documento PADRÕES E ALGORITMOS CRIPTOGRÁFICOS DA ICP-BRASIL (DOC-ICP-01.01).

§ 1º Todas as demais cláusulas do DOC-ICP-01.01, na sua versão 2.4, em sua ordem originária, integram a presente versão 2.5 e mantêm-se válidas.

§ 2º O documento referido no caput encontra-se disponibilizado, em sua totalidade, no sítio http://www.iti.gov.br.

Art. 3º Esta Instrução Normativa entra em vigor na data de sua publicação.

RENATO DA SILVEIRA MARTINI