Uma análise dos certificados digitais utilizados nas conexões TLS dos aplicativos de *Mobile banking* na plataforma Android

Diego Baierle Sebastiany
Mirelle Daiara Vieira Freitas
Luciano Ignaczak
Universidade do Vale do Rio dos Sinos - Unisinos

Resumo

Atualmente, é cada vez mais comum a utilização de aplicativos de *mobile banking* em smartphones. Tais aplicativos devem implementar o protocolo TLS para fornecer o sigilo e a autenticação necessária para proteger a comunicação entre o cliente e o seu banco. No entanto, como trabalhos nesta área já apresentaram, muitos aplicativos possuem problemas no código do TLS que comprometem a segurança da comunicação. Este artigo analisou se os aplicativos executam o protocolo TLS de forma correta, identificando se os certificados digitais usados por bancos de três países nas conexões TLS com as aplicações de m-banking seriam reconhecidos como confiáveis pela plataforma Android. Além disso, foi discutido o tamanho da chave e o período de validade desses certificados digitais, características essenciais para serem considerados resistentes contra ataques de quebra da sua chave criptográfica. Os resultados apresentaram um cenário preocupante para o segmento de aplicativos de m-banking, pois aproximadamente 31% dos aplicativos analisados utilizaram certificados digitais não confiáveis pela versão do Android avaliada e nenhum alerta de conexão insegura foi emitido durante os testes.
Palavras-chave: *mobile banking*, Android, certificado digital, TLS, dispositivos móveis

A crescente popularização da internet tem levado a um aumento expressivo da quantidade de dispositivos conectados. Com isso, cresceu também a quantidade de usuários que utilizam aplicativos para gerenciamento e movimentação financeira de suas contas bancárias. Uma pesquisa mostrou que 52% das transações bancárias feitas no Brasil em 2014 foram realizadas via internet e *mobile banking* (m-banking). Entre as contas ativas no país em 2014, 24% dos clientes (25 milhões) realizaram transações utilizando m-banking em seus smart hones (Febraban, 2015).

Os aplicativos de m-banking precisam implementar mecanismos de segurança para garantir que os dados do usuário não fiquem vulneráveis a roubo e interceptação na Internet. O protocolo TLS (Trans ort Layer Security) é utilizado como padrão para fornecer a segurança nesse ambiente (Elkhodr, Shahrestani & Kourouche, 2012). Além de garantir o sigilo e a integridade na comunicação, o protocolo TLS autentica o servidor do banco no qual o aplicativo está se conectando. A autenticação é importante e necessária para confirmar que o computador que está respondendo é realmente a entidade que afirma ser (Stallings, 2015; Adams & Lloyd, 2003).

O estabelecimento de uma comunicação segura exige a correta execução do protocolo TLS. Na execução do handshake do TLS, o banco envia o seu certificado digital para que o cliente (o aplicativo) verifique a sua identidade. Para isso, o aplicativo deve utilizar um dos certificados raiz instalados no sistema Android para fazer a validação da confiança do certificado digital do banco. A tentativa de conexão deveria falhar se o certificado raiz usado pelo banco não for considerado

DOI: 10.18256/2359-3539/reit-imed.v2n1p66-75

confiável pelo sistema Android. Esse é o processo correto para a realização da autenticação do servidor do banco. No entanto, muitas vezes os aplicativos falham ou simplesmente não executam a validação do certificado digital (Six, 2012). Como é apresentado na seção Trabalhos Relacionados, em muitos casos, tais falhas são decorrentes de erros no código do protocolo TLS desenvolvido para o aplicativo. Essas falhas invalidam o processo de autenticação e permitem que intrusos interceptem os dados manipulados pelos aplicativos de m-banking, possibilitando ações criminosas que resultam em prejuízos financeiros.

Além da validação da confiança no certificado digital do banco, o tamanho da chave criptográfica e o período de validade de um certificado digital são muito importantes e devem ser considerados no momento de sua emissão para conferir-lhe resistência contra ataques de força bruta. As recomendações mais recentes sugerem um tamanho de chave de no mínimo 2.048 bits (Barker & Roginsky, 2011). Quanto ao período de validade, recomenda-se que certificados digitais com tamanho da chave de 1.024 bits devem possuir até 1 ano de validade certificados digitais com tamanho de chave de 2.048 bits devem possuir no máximo 2 anos de validade e certificados digitais com tamanho de chave de 4.096 bits podem possuir validade de até 16 anos (OMeally, 2009).

O objetivo deste trabalho foi analisar a confiança dos certificados digitais utilizados por bancos nas conexões TLS com os aplicativos de m-banking na plataforma Android, além de outras duas características: o período de validade e o tamanho da chave criptográfica desses certificados. A análise foi realizada a partir de uma amostra de 60 aplicativos de m-banking disponibilizados por bancos de três países: Brasil, Estados Unidos e Reino Unido. Esses dois últimos foram escolhidos por figurarem, juntamente com o Brasil, entre as maiores economias mundiais em 2014 (Group, 2014) e por utilizarem a língua inglesa, proporcionando maior facilidade de interação com os aplicativos para a análise. Para cada aplicativo foi realizada uma simulação de acesso à conta bancária e, a partir do tráfego capturado, foi verificada a utilização do TLS e obtidos os certificados digitais utilizados na conexão. A principal contribuição deste artigo é oferecer uma análise sobre a segurança oferecida nas conexões TLS dos aplicativos de m-banking brasileiros, considerando as caraterísticas verificadas, e apresentar um comparativo com as aplicações de m-banking de outros dois países.

O restante deste trabalho segue com a seção que apresenta alguns trabalhos relacionados, e após essa, a seção que contém a fundamentação teórica utilizada como base para este trabalho. A seção Metodologia descreve a metodologia utilizada na realização desta análise a seção Resultados mostra os resultados obtidos da análise dos aplicativos de m-banking e, por fim, a última seção expõe as considerações finais deste trabalho.

Trabalhos Relacionados

Muitos aplicativos vêm apresentando problemas de implementação do protocolo TLS, os quais têm motivado muitos trabalhos que discutem suas causas e possíveis soluções. O trabalho de Georgiev et al. (2012) mostra que a segurança oferecida pelo TLS depende da correta validação do certificado digital fornecido quando a conexão é estabelecida. Esse trabalho analisou como alguns softwares e aplicativos implementam as funções do TLS para validação dos certificados digitais e mostrou que mesmo aplicativos desenvolvidos por grandes empresas possuem falhas graves. Muitas vezes, segundo os autores, as falhas na validação de um certificado é causada pela falta de entendimento e interpretação das APIs (A lication Programming Interface) utilizadas pelos desenvolvedores. A falta de conhecimento e informação sobre essas APIs conduz o desenvolvedor ao erro e deixa o aplicativo vulnerável a ataques do homem do meio. O trabalho de Hubbard, Weimer e Chen (2014) realizou uma pesquisa com o objetivo de identificar falhas na validação dos certificados digitais. Com uma amostra de 41 aplicativos para a plataforma Android, 11 falharam em estabelecer a relação de confiança necessária, pois aceitaram um certificado digital falsificado que, portanto, não pertencia à base de confiança do Android. O artigo também destacou que a falha dos aplicativos pode estar relacionada às APIs utilizadas.

Por serem pouco restritivas, permitem que os desenvolvedores cometam erros de implementação do código, permitindo que qualquer certificado digital seja aceito pelo aplicativo ou, até mesmo, que nenhuma validação seja realizada.

A inconsistência da base de certificados raiz confiáveis da plataforma Android também já foi alvo de estudo. Vallina-Rodriguez, Amann, Krei-

bich, Weaver e Paxson (2014) examinaram os certificados raiz instalados em diversas versões do Android, em vários dispositivos. O trabalho analisou a composição dessas bases de confiança e como elas variam de acordo com a versão do sistema e da marca do dispositivo. Como resultado, foi verificado que a base oficial de certificados digitais confiados pelo Android é modificada ou ampliada. Em alguns casos, o próprio fabricante do dispositivo e/ou a operadora de telefonia adicionam certificados digitais aos dispositivos para estabelecerem uma relação de confiança para aplicativos embarcados ou prestação de serviços. O trabalho também alertou para o fato de que, em dispositivos que rodam com usuário root, aplicativos maliciosos podem instalar certificados digitais no sistema sem o conhecimento do usuário, quebrando o modelo de confiança de certificados digitais supervisionados e auditados como confiáveis.

Fahl et al. (2012) investigaram o uso inadequado do TLS em 13.500 aplicativos de diversas categorias, obtidos da Google Play Market. Dos aplicativos analisados, 1.074 (17,28% dos que utilizaram o TLS) continham erros de código que permitiam a validação de qualquer certificado digital ou confiavam em qualquer certificado raiz. Os autores mostraram também que as falhas na implementação do TLS ocorrem porque o Android permite que os desenvolvedores criem códigos personalizados para seus aplicativos. Eles destacaram que esse recurso devia ser desativado e que as APIs para Android deviam forçar a utilização das implementações padrão do TLS. Em outro trabalho, Fahl, Harbach, Perl, Koetter e Smith (2013) continuaram investigando as possíveis causas da má implementação do TLS em aplicativos. O resultado da pesquisa mostrou que as causas não são simplesmente a falta de cuidado por parte dos desenvolvedores, mas também questões e limitações envolvendo o atual paradigma de desenvolvimento do TLS. O trabalho sugeriu mudanças no atual paradigma em direção a uma maior abstração do código fornecido pelas APIs, permitindo que desenvolvedores utilizem corretamente o TLS com menos esforço e prevenindo falhas na validação dos certificados digitais.

Os trabalhos relacionados reforçam a necessidade de um maior cuidado no desenvolvimento do código do protocolo TLS em aplicativos que transmitem dados confidenciais. Os artigos citados nesta seção realizaram análises dos certificados digitais utilizados no TLS em diversos aplicativos, sem abordar um segmento específico. Já este artigo, analisou especificamente o segmento bancário, notadamente operando com dados confidenciais dos correntistas, buscando verificar a confiança atribuída pelos aplicativos de m-banking aos certificados digitais dos bancos.

Fundamentação Teórica

Esta seção apresenta a fundamentação teórica para o desenvolvimento deste trabalho, descrevendo brevemente características e funções dos aplicativos de m-banking, do protocolo TLS, dos certificados digitais, e do gerenciamento de certificados digitais no sistema Android.

Mobile banking

Segundo Pousttchi e Schurig (2004), m-banking pode ser definido como: "a execução de serviços financeiros, no decurso dos quais - no âmbito de um procedimento eletrônico - o cliente utiliza técnicas de comunicação móvel em conjunto com dispositivos móveis". Outra definição é fornecida em uma pesquisa do Federal Reserve System - FED (dos Estados Unidos) que define m-banking como o acesso à conta bancária realizada a partir de aplicativos instalados no telefone celular, ou através da página Web do banco utilizando o navegador Web do dispositivo móvel, ou ainda via mensagens de texto (Federal Reserve System, 2014).

De fato, os aplicativos de m-banking possibilitam que clientes de bancos utilizem seus dispositivos móveis, como smart hones e tablets, para verificarem o saldo de suas contas bancárias, realizarem pagamentos, transferências, investimentos, e demais transações bancárias. Para os clientes, tais aplicativos proporcionam maior comodidade e segurança, pois assim podem utilizar os serviços do banco sem a necessidade de deslocamento até uma agência física (Pousttchi & Schurig, 2004).

Devido à natureza sensível dos dados manipulados, uma preocupação inerente aos aplicativos de m-banking deve ser a segurança na comunicação que o aplicativo estabelece entre o cliente e o banco. A pesquisa de Federal Reserve System (2014) mostrou que o maior motivo para a não utilização de serviços bancários móveis é justamente a preocupação com aspectos de segurança da tecnologia. Entretanto, a confidencialidade,

integridade, e autenticidade na comunicação pode ser alcançada com o desenvolvimento correto do protocolo TLS no código do aplicativo (Six, 2012 Thomas, 2000).

Transport Layer Security - TLS

O protocolo TLS foi definido pela Internet Engineering Task Force (IETF) em 1999 como um padrão de Internet para o fornecimento de segurança na comunicação de dados. Atualmente, ele está definido em sua versão 1.2 na RFC 5246, a qual menciona que "o protocolo permite que aplicações cliente/servidor se comuniquem de um modo que é concebido para evitar a espionagem, adulteração ou a falsificação de mensagens" (Dierks & Rescorla, 2008). O padrão define que a comunicação sempre deve ser iniciada pelo cliente. O protocolo TLS emprega criptografia para encriptar a comunicação entre cliente e servidor, e também para autenticar ambas as partes. O processo de autenticação garante que o servidor e/ou o cliente são realmente quem afirmam ser (Stallings, 2015).

O TLS possui um subprotocolo chamado protocolo de estabelecimento de sessão, ou protocolo de handshake. O protocolo de handshake é responsável pela negociação de parâmetros para a definição dos algoritmos criptográficos que serão utilizados em uma sessão que o cliente inicia com o servidor. Além disso, é o protocolo de handshake que realiza a autenticação das partes com base em certificados digitais. Embora ambas as partes possam ser autenticadas, o usual é apenas a autenticação do servidor (Thomas, 2000). O estabelecimento de uma nova sessão envolve a troca de nove mensagens como mostra a figura 1, na qual a numeração indica a ordem de envio, e o sentido é indicado pela direção da seta. Cada mensagem possui uma função distinta. A mensagem ClientHello (1) é enviada pelo cliente para iniciar a negociação de uma nova sessão com o servidor a mensagem ServerHello (2) é enviada pelo servidor com as definições iniciais da sessão, após a qual o servidor envia a mensagem Certificate (3) que contém o certificado digital que o cliente utilizará para autenticar o servidor, e a mensagem ServerHelloDone (4) que sinaliza para o cliente o fim dessa fase da negociação o cliente envia a mensagem ClientKeyExchange (5) que é encriptada com a chave pública do servidor (que foi fornecida através do certificado digital na mensagem 3) a mensagem ChangeCipherSpec (6) é enviada ao servidor para ativar as especificações negociadas para todas as futuras mensagens, e uma mensagem Finished (7) indica o fim dessa fase da negociação permitindo ao servidor verificar as especificações da sessão da mesma forma uma mensagem ChangeCipherSpec (8) é enviada para o cliente confirmando as especificações negociadas para todas as futuras mensagens, e uma mensagem Finished (9) permite ao cliente verificar as especificações da sessão (Thomas, 2000). O protocolo de handshake possui outras mensagens que não são mostradas na figura 1. Essas mensagens são opcionais ou dependentes da finalidade da sessão envolvida (Thomas, 2000 Stallings, 2015).

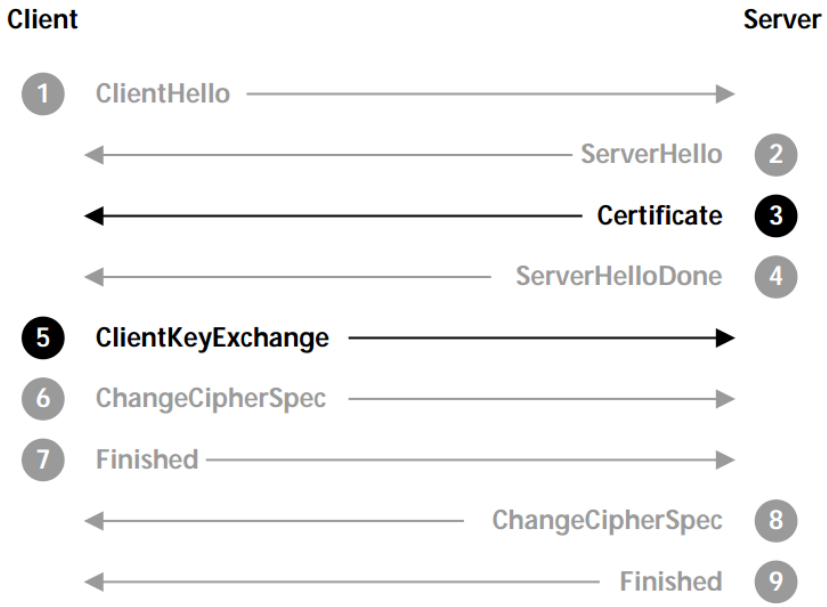

Figura 1. Mensagens do *handshake*. Adaptado de (Thomas, 2000).

A autenticação do servidor é garantida por duas mensagens em particular, a mensagem Certificate (3) e a mensagem ClientKeyExchange (5). O processo de autenticação do servidor é realizado pelo cliente utilizando o certificado digital fornecido na mensagem Certificate (3). Essa mensagem fornece, na realidade, toda a cadeia de certificados digitais do caminho de certificação cuja raiz é uma autoridade de certificação. Cabe ao cliente verificar se ele pode confiar no certificado digital do servidor, o que faz verificando se a autoridade de certificação é uma das confiadas pelo sistema operacional. Além de estabelecer a confiança, o cliente verifica se os dados do certificado digital (como o nome de domínio, por exemplo) conferem com os do servidor. A mensagem ClientKeyExchange (5), por sua vez, fornece uma confirmação implícita para o processo de autenticação. A confirmação ocorre porque o cliente encripta a mensagem ClientKeyExchange (5) com a chave pública do servidor. Dessa forma, apenas o servidor pode decifrar a mensagem para continuar a comunicação com o cliente, pois apenas o servidor possui a chave privada correspondente à chave pública utilizada pelo cliente (Thomas, 2000).

Certificados Digitais

Certificados digitais são utilizados em conjunto com a criptografia de chave pública (criptografia assimétrica), que utiliza um par de chaves criptográficas diferentes que são matematicamente relacionadas. A operação que uma chave realiza (cifragem ou decifragem) pode ser desfeita somente pela chave correspondente a este par de chaves criptográficas (Stallings, 2015). Assim, uma das chaves (a chamada chave privada) deve ser conhecida apenas pela entidade que a possui. A outra chave (que é chamada de chave pública) pode ser disponibilizada publicamente para qualquer pessoa que deseje utilizar meios criptográficos de comunicação com o titular da chave pública. Entretanto, os esquemas de criptografia de chave pública são seguros apenas se a autenticação da chave pública for garantida (Stallings, 2015), pois ela em si mesma não fornece nenhum meio para a identificação do seu titular. Para esse fim, é utilizado um esquema de certificação da chave pública, cujo produto é o certificado digital. O certificado digital X.509 fornece as informações necessárias para relacionar uma entidade como titular exclusivo de uma chave pública (Adams & Lloyd, 2003).

A criptografia assimétrica permitiu a execução de serviços que antes não eram possíveis, ou eram limitados, devido às características da criptografia simétrica. Entre esses serviços está a assinatura digital. A assinatura digital funciona de forma análoga à assinatura escrita de uma pessoa, de forma que uma única entidade pode assinar um dado ou informação, mas qualquer número de entidades pode ler essa assinatura e verificar a sua autenticidade. A assinatura digital é realizada com a utilização da chave privada da entidade que assina. A autenticidade dessa assinatura é garantida pelo fato de que somente essa entidade possui a chave privada. Utilizando a chave pública correspondente, qualquer outra entidade pode verificar que essa assinatura é autentica (Adams & Lloyd, 2003).

A fim de desenvolver um sistema capaz de gerenciar e fornecer certificados digitais de forma segura, foi criada uma infra-estrutura de chave pública (PKI) para a Internet, que está definida na RFC 5280 (Cooper et al., 2008). O objetivo principal para o desenvolvimento de uma PKI é permitir a aquisição segura, conveniente e eficiente de chaves públicas (Adams & Lloyd, 2003). A principal entidade no esquema de uma PKI é a autoridade de certificação (AC), que é a raiz no caminho de certificação de certificados digitais. Ela é um instrumento íntegro, acreditado e reconhecido como confiável por outras entidades, responsável pela emissão de certificados digitais em conformidade com a política de certificação ditada por uma autoridade. A AC utiliza o serviço de assinatura digital para autenticar os dados do certificado digital que emitiu, e assim relacionar uma chave pública à entidade titular desse certificado digital (Adams & Lloyd, 2003). Esse processo é ilustrado na figura 2. De posse do certificado digital da AC que assinou um determinado certificado digital, qualquer entidade pode verificar a autenticidade dessa assinatura e, assim, verificar se este certificado digital pode ser confiado como legítimo. Um erro nessa verificação indica que o certificado digital não pode ser considerado confiável, pois possivelmente foi adulterado ou falsificado por outra entidade (Stallings, 2015).

A ITU-T (2012) descreve as normas para a padronização dos certificado digitais X.509. Atualmente, os certificados digitais estão na versão 3, e possuem a estrutura mostrada na figura

3. Os dados de todos os campos são utilizados como entrada em um algoritmo criptográfico juntamente com a chave privada de uma AC para produzir uma assinatura digital, que é concatenada como um último campo na estrutura do certificado digital X.509.

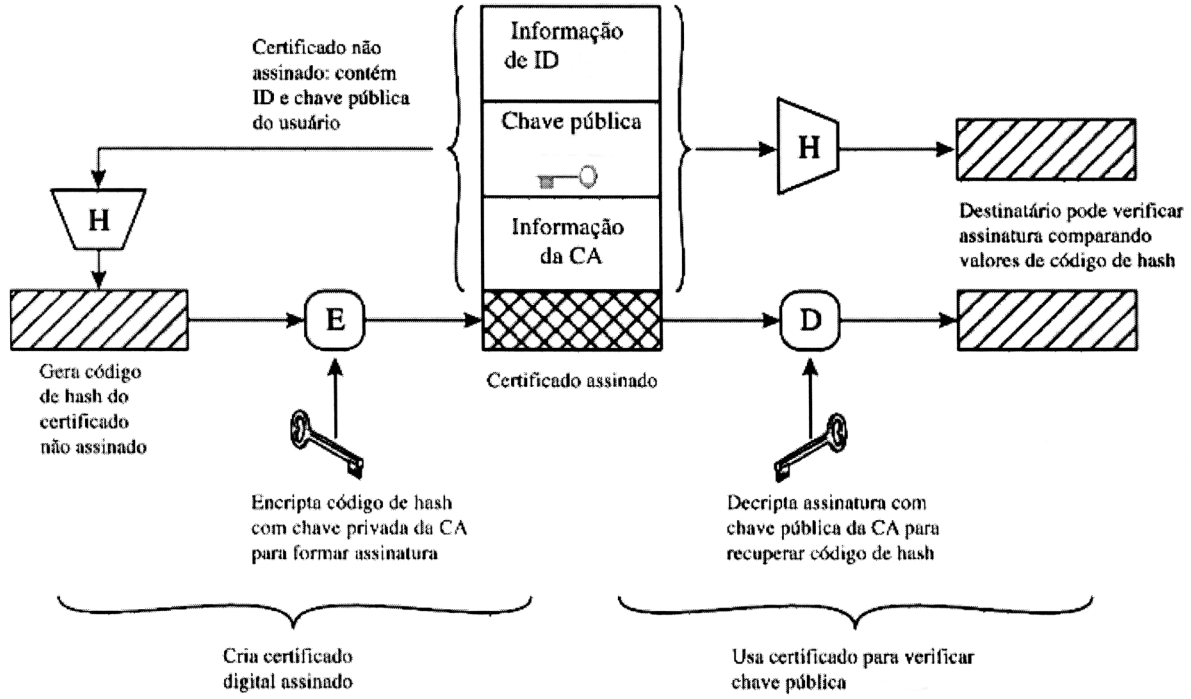

Figura 2. Uso do certificado da chave pública. Adaptado de (Stalling, 2015).

Gerenciamento de Certificados Digitais na plataforma Android

A plataforma Android utiliza a API Java Secure Socket Extension (JSSE) para estabelecer classes e interfaces necessárias para o funcionamento do SSL e TLS. Utilizando-se dessa estrutura, o sistema Android consegue usar os certificados digitais X.509 para fornecer autenticação e confidencialidade para a comunicação (Android, 2015b).

Como é possível criar seu próprio par de chaves e certificado digital, o sistema Android mantém um repositório que contém os certificados digitais das autoridades de certificação confiadas pela plataforma. Essa lista é considerada como confiável pelo sistema e é utilizada pela interface TrustManager que realiza as decisões de confiança baseada nesse repositório. A lista de certificados digitais é atualizada a cada nova atualização da plataforma, desde a versão 4.2 (Jelly Bean) do Android (Android, 2015c). O repositório com as ACs confiadas pelo sistema Android é recuperado pela classe Keystore, que também é conhecida como Truststore (Android, 2015d).

É possível adicionar exceções com relação a este repositório de ACs confiáveis e tornar confiável um certificado digital não reconhecido até então como confiável pela plataforma Android. Para isso, uma nova Keystore deve ser criada e utilizada para iniciar um TrustManager, que é responsável pela tomada de decisão no sistema e decide qual certificado digital é ou não válido, com base em uma consulta ao repositório de ACs especificado (Android, 2015c).

Metodologia

Para a realização desta análise foi selecionada uma amostra com 60 aplicativos de m-banking divididos igualmente em três países: Brasil, Estados Unidos (EUA) e Reino Unido (UK). A seleção dos aplicativos foi realizada utilizando rankings do Banco Central do Brasil[1], do Federal Reserve System[2] para os EUA, e do Relbanks[3] para o UK, que classificam os bancos com maiores ativos em cada país. Baseados nestes rankings, os autores selecionaram os 20 primeiros bancos de varejo que possuem aplicativos de m-banking. A lista da

1 Disponível em: http://www4.bcb.gov.br/top50/port/top50.asp
2 Disponível em: http://www.federalreserve.gov/Releases/Lbr/current/default.htm
3 Disponível em: http://www.relbanks.com/europe/uk

Relbanks possui apenas 11 bancos e foi utilizada porque não foi encontrado um ranking oficial do Banco Central do Reino Unido. A amostra de aplicativos do país foi incrementada com mais 10 bancos conhecidos do Reino Unido, retirados do site do seu Banco Central[4]. A análise consistiu na avaliação das seguintes características de cada aplicativo:

se o aplicativo utiliza o protocolo TLS para comunicação segura;

a verificação da confiança no certificado digital do banco;

o período de validade do certificado digital do banco;

o tamanho da chave do certificado digital do banco;

O *software* Genymotion[5] foi usado para emular um dispositivo rodando a versão 4.4 do sistema Android, que está instalada atualmente em 39,3% dos dispositivos dessa plataforma (Android, 2015a). Desse dispositivo foram extraído todos os certificados digitais armazenados em /system/etc/security/cacerts/. Esses são os certificados digitais das autoridades de certificação confiadas por esta versão do Android. Os certificados digitais extraídos foram armazenados para, posteriormente, analisar a confiança dos certificados raiz utilizados nas conexões TLS pelos aplicativos de m-banking. Para possibilitar a análise, os aplicativos de m-banking selecionados foram instalados no dispositivo virtual. Além disso, para que fosse possível a captura do tráfego TLS gerado pelo aplicativo de m-banking foi utilizado o tcpdump, disponível no emulador.

Após a instalação de cada aplicativo de m-banking, foram realizadas tentativas de acesso à conta bancária. O acesso foi simulado pela inserção dos dados necessários (como número da conta e senha) aceitos pelo aplicativo, para que ele iniciasse a comunicação com o banco, e assim, estabelecesse a conexão segura (TLS). Com o tráfego gerado pala simulação do acesso, foi avaliado o primeiro critério desta análise: se o aplicativo utiliza o TLS.

No caso dos aplicativos de m-banking que possibilitaram a verificação da implementação do protocolo TLS, da captura do tráfego foram extraídos os certificados digitais utilizados pelo handshake: o certificado do banco e o certificado raiz, que é utilizado para avaliar a relação de confiança entre o banco e o sistema Android. A partir do certificado digital do banco foi avaliado o tamanho da chave criptográfica bem como o seu período de validade. Para auxiliar na consolidação dos resultados dessa análise foi utilizado um software desenvolvido pelos autores, na linguagem C#, que coletou os dados dos certificados e exportou os resultados no formato XML. O arquivo exportado foi utilizado como fonte para a construção de uma planilha.

Um segundo software na linguagem C# também necessitou ser desenvolvido pelos autores para analisar a confiança dos certificados digitais raiz capturados. Esse software realizou o cruzamento entre os certificados digitais raiz capturados e a base de certificados digitais raiz considerada confiável pela versão avaliada do Android. O cruzamento desses certificados digitais consistiu em comparar os campos Subject Key Identifier, ou na ausência deste, a própria chave pública contida no campo Subject Public Key Info. A saída desse programa foi salva e adicionada à planilha anterior, usada como base para a avaliação dos resultados.

Não foi possível avaliar alguns aplicativos de m-banking pois a captura do tráfego desses aplicativos no momento da autenticação não apresentou a utilização do TLS, tampouco revelou os dados do usuário em texto claro. Isso pode acontecer quando o aplicativo implementa os requisitos de segurança na camada de aplicação. Por isso, não é possível afirmar que o aplicativo falha em oferecer segurança para o usuário. Os aplicativos com essas características foram classificados como indefinidos.

A última etapa desse trabalho consistiu na realização da análise dos resultados obtidos. Nesta etapa foram efetuados cálculos de porcentagem, cruzamento de informações e médias, a fim de comparar as definições dos bancos dos três países em relação aos dados dos certificados digitais que são alvo deste artigo.

Resultados

A análise dos 60 aplicativos selecionados resultou em 2 aplicativos classificados como indefi-

4 Disponível em: http://www.bankofengland.co.uk
5 Disponível em: https://www.genymotion.com

nidos, ambos do Brasil, e em 58 aplicativos que utilizaram o TLS para estabelecer a conexão segura.

Não foi possível verificar a utilização do TLS ao analisar a captura do tráfego gerado pelos 2 aplicativos que foram classificados como indefinidos. Embora não seja possível afirmar, o mecanismo de segurança utilizado por esses aplicativos pode ser o próprio TLS, mas implementado de forma personalizada pelos desenvolvedores. Isso é possível porque as APIs utilizadas para o Android permitem esse nível de personalização do código.

O resultado mostrou que os outros 58 aplicativos analisados utilizaram o TLS, realizando o handshake e apresentando o certificado digital do banco como é padrão do protocolo. Porém, 18 (31%) desses aplicativos não poderiam ser considerados confiáveis, pois esses utilizam certificados digitais emitidos por autoridades de certificação que não são confiadas pelo Android. O resultado dessa verificação é apresentado na Tabela 1.

Tabela 1. Certificados raiz sem relação de confiança com o Android

Origem	Total de certificados raiz não confiáveis	Percentual de certificados raiz não confiáveis
Brasil	8	44,44%
Estados Unidos	7	35,00%
Reino Unido	3	15,00%
TOTAL	18	31,03%

A análise da relação de confiança mostrou que o cenário mais preocupante é o brasileiro, onde 44% dos aplicativos analisados não podem ser considerados confiáveis pela versão da plataforma Android analisada. O Reino Unido apresentou o menor número de certificados digitais não confiáveis (15%), porém, ainda é preocupante considerando que o segmento analisado é o bancário, que deveria possuir um cuidado adicional no uso de certificados digitais.

Como foi mostrado pelos trabalhos relacionados, as falhas de validação da confiança expõem o cliente a diversos riscos, e são resultado da forma de implementação do código do TLS do aplicativo. Semelhante às análises nesses trabalhos, esta análise dos aplicativos de m-banking revelou um cenário inquietante, pois nenhum dos aplicativos que utilizam certificados não confiados pela plataforma Android apresentou qualquer mensagem de alerta durante o handshake do TLS, para informar o usuário que a conexão não é segura.

A segunda parte desta análise, avaliou o período de validade e o tamanho da chave criptográfica do certificado digital do banco. Todos os 58 certificados digitais possuem o tamanho da chave igual a 2.048 bits com períodos de validade distintos. Os períodos de validade dos certificados digitais usados pelos aplicativos de m-banking são apresentados na Tabela 2.

Tabela 2. Período de validade dos certificados digitais dos bancos

Origem	Total de certificados	Perído de validade			
		1 ano	2 anos	3 anos	4 anos
Brasil	18	10	8	0	0
Estados Unidos	20	14	3	1	2
Reino Unido	20	12	8	0	0

Embora todos os certificados digitais dos bancos analisados atendam à recomendação do NIST no que diz respeito ao tamanho da chave (Barker & Roginsky, 2011), 3 deles, todos dos EUA, possuem o período de validade superior a 2 anos. Conforme a recomendação da Microsoft, o período máximo de validade deve ser de 2 anos para certificados com tamanhos de chave de 2.048 bits. Um período de validade muito grande diminui a resistência da chave associada ao certificado digital, pois os avanços da tecnologia de computação podem comprometer um certificado digital que, para os padrões atuais, é considerado forte.

Considerações Finais

Quando o usuário instala e utiliza um aplicativo em seu smart hone, ele o faz confiando que a comunicação e seus dados estarão seguros. Quando se trata do segmento de m-banking, espera-se que todos os aplicativos implementem correta-

mente o protocolo TLS para atender os requisitos de segurança e proteger o usuário. Ao usuário resta apenas confiar no aplicativo, pois o sistema Android não oferece nenhuma indicação de que a comunicação é estabelecida de forma segura.

Existem normas e recomendações que os desenvolvedores de aplicativos devem seguir para atender requisitos no desenvolvimento de seus aplicativos e evitar erros comuns ao utilizar códigos personalizados. O segmento de m-banking deve observar especialmente as recomendações de segurança como a do NIST (Barker & Roginsky, 2011) que especifica o tamanho mínimo da chave do certificado digital em 2.048 bits. Como foi mostrado nos resultados deste trabalho, todos os aplicativos seguiram essa recomendação pois todos possuem tamanho da chave igual a 2.048 bits. No entanto, 3 desses certificados digitais possuem o período de validade maior que 2 anos, em desacordo com a recomendação da Microsoft (OMeally, 2009). Isso pode resultar em uma falha de segurança, pois os avanços da tecnologia de computação poderão permitir a quebra de chaves com tamanho de 2.048 bits durante o período de validade do certificado digital.

Ademais, uma parcela significativa dos aplicativos, considerando-se sistemas de m-banking, falham na implementação da validação do certificado digital porque a relação de confiança que deveria existir entre o certificado raiz do banco e o sistema Android não é estabelecida. Sem a validação da confiança, um certificado digital é aceito sem qualquer restrição, quebrando completamente o sistema de certificação digital, supervisionado e auditado como confiável. Esse problema mostra-se ainda mais grave quando considerado que isso ocorre de forma transparente para o usuário. Embora o protocolo TLS forneça o recurso para avisar o usuário que a relação de confiança não foi estabelecida, muitas vezes esse recurso é desativado ou mau implementado pelo desenvolvedor. Neste trabalho, dos 18 aplicativos que falharam ao estabelecer a relação de confiança, nenhum mostrou qualquer mensagem de aviso sobre essa falha, e todos prosseguiram funcionando como se nenhum erro tivesse ocorrido.

Este trabalho analisou uma amostra reduzida de certificados digitais utilizados por aplicativos de m-banking durante a conexão TLS. Como trabalho futuro é sugerido a análise de uma amostra mais ampla que reflita com mais precisão a realidade no segmento dos aplicativos de m-banking. Além disso, trabalhos futuros podem comparar as características de certificados digitais usados no TLS por aplicativos de outros segmentos.

Referências

Adams, C. & Lloyd, S. (2003). Understanding pki: concepts, standards, and deployment considerations (Second). Boston, MA: Addison-Wesley Pearson Education.

Android, D. (2015a). Dashboards: Platform versions. Recuperado de https://developer.android.com/about/dashboards/index.html

Android, D. (2015b). Develop reference: Package javax.net.ssl. Recuperado de http://developer.android.com/reference/javax/net/ssl/package-summary.html

Android, D. (2015c). Develop training: Security with http and ssl. Recuperado de https://developer.android.com/training/articles/security-ssl.html

Android, D. (2015d). Develop training: Android keystore system. Recuperado de https://developer.android.com/intl/pt-br/training/articles/keystore.html

Barker, E. & Roginsky, A. (2011). Transitions: recommendation for transitioning the use of cryptographic algorithms and key lengths. NIST Special Publication 800-131A. Recuperado de http://csrc.nist.gov/publications/nistpubs/800-131A/sp800-131A.pdf

Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R. & Polk, W. (2008). Internet x.509 public key infrastructure certi_cate and certi_cate revocation list (crl) pro_le (RFC No 5280). Recuperado de https://datatracker.ietf.org/doc/rfc5280

Dierks, T. & Rescorla, E. (2008). The transport layer security (tls) protocol version 1.2 (RFC No 5246). Recuperado de www.rfc-editor.org/info/rfc5246

Elkhodr, M., Shahrestani, S. & Kourouche, K. (2012). A proposal to improve the security of *mobile banking* applications. Em Ict and knowledge engineering (ict knowledge engineering), 2012 10th international conference on (pp. 260_265).

Fahl, S., Harbach, M., Muders, T., Baumgärtner, L., Freisleben, B. & Smith, M. (2012). Why eve and mallory love android: an analysis of android ssl (in)security. Em Proceedings of the 2012 acm conference on computer and communications security (pp. 50_61). CCS '12. Raleigh, North Carolina, USA: ACM. Recuperado de http://doi.acm.org/10.1145/2382196.2382205

Fahl, S., Harbach, M., Perl, H., Koetter, M. & Smith, M. (2013). Rethinking ssl development in an appi_ed world. Em Proceedings of the 2013 acm sigsac conference on computer & commu-

nications security (pp. 49_60). CCS '13. Berlin, Germany: ACM. Recuperado de http://doi.acm.org/10.1145/2508859.2516655

Febraban. (2015). Pesquisa febraban de tecnologia bancária 2014. Recuperado de https://www.febraban.org.br/Noticias1.asp?id_texto=2626

Federal Reserve System, F. (2014). Consumers and mobile _nancial services _ march 2015. Recuperado de http://www.federalreserve.gov/econresdata/mobile-devices/2015-preface.htm

Georgiev, M., Iyengar, S., Jana, S., Anubhai, R., Boneh, D. & Shmatikov, V. (2012). The most dangerous code in the world: validating ssl certi_cates in non-browser software. Em Proceedings of the 2012 acm conference on computer and communications security (pp. 38_49). CCS '12. Raleigh, North Carolina, USA: ACM. Recuperado de http://doi.acm.org/10.1145/2382196.2382204

Group, W. B. (2014). World development indicators: Gdp ranking. Recuperado de http://data.worldbank.org/data-catalog/GDP-ranking-table

Hubbard, J., Weimer, K. & Chen, Y. (2014). A study of ssl proxy attacks on android and ios mobile applications. Em Consumer communications and networking conference (ccnc), 2014 ieee 11th (pp. 86_91).

ITU-T, I. T. U. T. S. S. (2012). Information technology - open systems interconnection - the directory: public-key and attribute certi_cate frameworks: Recommendation itu-t x.509. Recuperado de http://www.itu.int/rec/T-REC-X.509-201210-I

OMeally, Y. (2009). Recommendations for pki key lengths and validity periods with con_guration manager. Recuperado de http://blogs.technet.com/b/con_gmgrteam/archive/2009/06/12/recommendationsfor-pki-key-lengths-and-validity-periods-with-con_guration-manager.aspx

Pousttchi, K. & Schurig, M. (2004). Assessment of today's *mobile banking* applications from the view of customer requirements. Em Proceedings of the proceedings of the 37th annual hawaii international conference on system sciences (hicss'04) - track 7 - volume 7 (pp. 70184.1_). HICSS '04. Washington, DC, USA: IEEE Computer Society. Recuperado de http://dl.acm.org/citation.cfm?id=962755.963161

Six, J. (2012). Segurança de aplicativos android. São Paulo, SP: Novatec Editora Ltda.

Stallings, W. (2015). Criptogra_a e segurana de redes: princípios e práticas (Sixth). São Paulo, SP: Pearson Education do Brasil Ltda.

Thomas, S. (2000). Ssl & tls essentials: securing the web. Wiley.

Vallina-Rodriguez, N., Amann, J., Kreibich, C., Weaver, N. & Paxson, V. (2014). A tangled mass: the android root certi_cate stores. Em Proceedings of the 10th acm international on conference on emerging networking experiments and technologies (pp. 141_148). CoNEXT '14. Sydney, Australia: ACM. Recuperado de http://doi.acm.org/10.1145/2674005.2675015

Abstract

Currently, it is increasingly common to use *mobile banking* applications on smartphones. Such applications must implement the TLS protocol to provide confidentiality and authentication required for secure communication between the customer and his bank. However, as works in this area have been already submitted, many applications have problems in the TLS code that compromise the security of communication. This article examined if the TLS are running correctly by applications, identifying if the digital certificates used by banks in three countries on the TLS connections with m-banking applications would be recognized as reliable by the Android platform. Furthermore, has been discussed the key size and the period of validity of digital certificates, as characteristics that should be considered resistant against breakage attacks to its cryptographic key. The results showed a worrying scenario for the segment of m-banking applications because about 31% of the examined applications have used untrusted digital certificates by Android version that was evaluated, and any unsafe connection warning was issued during the tests.
Keywords: *mobile banking*, Android, digital certificates, TLS, mobile devices.