UNIVERSITY OF BEIRA INTERIOR
Engineering

Secure Authentication Mechanisms for the Management Interface in Cloud Computing Environments

Liliana Filipa Baptista Soares

Dissertation Submitted in Partial Fulfillment of the Requirement for the Degree of Master of Science in
Computer Science and Engineering
(2nd Cycle Studies)

Supervised by Prof. Dr. Pedro Ricardo Morais Inácio

Covilhã, October 2013

To my beloved family,
especially to my grandparents.

Acknowledgements

The endurance to overcome obstacles or predisposed objectives in life depends on how each one envisages them. It is up to each one to extract the most valuable lessons from each journey and transform them into knowledge. Without the guidance and support of very particular persons, it would have not been possible to accomplish this dissertation and many lessons would have not been learned.

I hereby thank my mother and father, to whom I am mostly grateful, for helping me to achieve my goals, particularly those of my academic quest. I would like to thank my sister, Cátia, for her immeasurable support, and my grandparents, for always demonstrating their love and will to live.

I am also grateful to my supervisor, Prof. Dr. Pedro Ricardo Morais Inácio, not only for the help, charisma and readiness he demonstrated to complete this stage, but also for being resourceful in aiding throughout other steps of my professional life. I also acknowledge the Multimedia Signal Processing – Covilhã Group at the *Instituto de Telecomunicações*, which hosted this dissertation work and the final project of my first degree, and the research folk in it, particularly Miguel Neto for lending me a smartcard reader temporarily.

Last, but not least, I acknowledge my friends for their strength. Foremost, I would like to express my gratitude to Diogo, who always knew where and how to be present in the most important moments of my life in recent years. Thank you all.

Liliana Filipa Baptista Soares

“Proving that I am right would be admitting that I could be wrong.”

Pierre-Augustin de Beaumarchais (1732–1799)

Resumo

Nos últimos anos, a expressão *computação na nuvem* tem sido tema de discussão. A indústria adotou esta tecnologia massivamente enquanto que a academia se está a focar em melhorá-la, e a qual tem vindo a evoluir rapidamente. O paradigma da computação na nuvem consiste em adotar soluções de fornecedores de computação na nuvem, que são hospedadas nalgum centro de dados. Os clientes tornam-se, portanto, dependentes dessas (terceiras) entidades, já que estas ficam envolvidas nos seus negócios ao serem responsáveis pelas tecnologias de informação hospedadas em uma ou mais nuvens. Isto implica que os clientes migrem as suas infraestruturas de tecnologias de informação e serviços locais para essas nuvens, localizadas fora das suas instalações. Os serviços e infraestruturas que podem ser deslocalizados incluem, mas não estão limitados a, serviços de e-mail, aplicações web, bases de dados, ou servidores completos que passam a ser encapsulados em serviços acedidos via Internet. Como vantagens, as nuvens oferecem capacidades de armazenamento, processamento e rede de uma forma escalável e elástica, à medida que o cliente quer e de uma maneira automática, ao seguir o modelo de negócio de pagamento por utilização. Isto beneficia bastante os clientes, permitindo-lhes que se foquem nos seus negócios, abstraindo-os das infraestruturas e problemas relacionados com tecnologias de informação.

Os serviços fornecidos pelas nuvens encapsulam-se normalmente num dos três principais modelos de fornecimento seguintes: Infraestrutura-como-um-Serviço (IaaS), Plataforma-como-um--Serviço (PaaS), e *Software*-como-um-Serviço (SaaS). Estes são os alicerces para soluções de Qualquer-coisa-como-um-Serviço (XaaS) que podem ser customizadas consoante os requisitos dos clientes. IaaS mistura técnicas novas de virtualização com tecnologias atuais que permite correr sistemas operativos ou mesmo construir centros de dados virtuais. PaaS permite desenvolver aplicações de uma forma consistente através das plataformas da nuvem para correrem sobre elas, enquanto que o SaaS permite usufruir de *software* previamente desenvolvido, tendo o cliente pouco controlo sobre o mesmo. Estes modelos podem ser implementados sobre nuvens privadas ou públicas, ou então sobre uma versão híbrida das duas. O modelo público significa que os serviços subscritos são acedidos através da Internet e a partir de qualquer ponto do globo. As nuvens IaaS têm tipicamente uma interface de gestão para controlo de Máquinas Virtuais (VMs) e para organizar um centro de dados virtual, enquanto que as VMs são acedidas por protocolos de acesso remoto. A autenticação para estas interfaces de gestão é, portanto, de grande importância, visto que estão expostas aos perigos da Internet, contrariamente às ferramentas tradicionais de gestão que são colocadas dentro do perímetro de confiança de uma empresa em redes convencionais. Esta dissertação identifica em primeiro lugar tais problemas de segurança ao rever as abordagens existentes de autenticação, mostrando as suas vantagens e fraquezas. Por exemplo, uma única conta comprometida constitui um risco muito maior quando comparado a contas tradicionais de páginas web, porque os atacantes ganham controlo sobre

as virtuais e potencialmente também sobre configurações de segurança. Isto pode resultar em perdas de dados e de dinheiro para ambos os clientes e os fornecedores de computação na nuvem, dado que um atacante malicioso consegue terminar instâncias de virtuais que possam correr aplicações criticas para o negócio do cliente.

O problema também reside no facto de que autenticação por meio de palavra-passe estática tem vindo, ao longo do tempo, a ficar mais degradada, sendo hoje em dia considerada insegura para utilização. As inúmeras fugas de dados do passado forneceram um conhecimento chave para compreender os hábitos dos utilizadores na escolha de palavras-passe, permitindo a atacantes compilar grandes listas de palavras-passe e desenvolver algoritmos eficientes para descobrir palavras-passe. As ameaças inerentes à utilização da Internet para aceder à interface de administração também agravam o problema. O ciberespaço é cada vez mais utilizado de forma abusiva e violenta com a ciberguerra, a cibercriminalidade, e a vigilância em massa afetando todos os utilizadores. Por exemplo, os autores de *malware* estão a focar-se nas plataformas móveis já que é para elas que a tecnologia evoluiu e as quais os utilizadores utilizam largamente. Como resposta, a indústria e a academia viraram as suas atenções para esquemas de autenticação alternativos baseados em Sistemas de *Login* Único (SSO) e Autenticação por Múltiplos Fatores (MFA). O SSO vem aliviar o fardo da gestão de múltiplas credenciais para aplicações na nuvem, enquanto que o MFA adiciona camadas adicionais de segurança em troca de uma quebra na usabilidade e um aumento no custo. Estas novas abordagens implementam esquemas de autenticação que recorrem a mecanismos da criptografia assimétrica e simétrica, por exemplo, e às novas tecnologias, sobretudo associadas ao móvel. A literatura contém várias contribuições nesta linha de investigação. No entanto, estas ainda não prestam total importância às infraestruturas inerentes e às ameaças atuais.

O âmbito desta dissertação está restrito aos tópicos de computação na nuvem e de autenticação na interface de administração de um cliente, estudando como este novo modelo de computação funciona, particularmente com ênfase na segurança, e revendo métodos de autenticação de forma ortogonal ao modelo de computação na nuvem e à perspetiva da segurança. Dado que as interfaces de gestão da nuvem são inerentemente mais arriscadas por estarem expostas e partindo do estudo inicial, esta dissertação propõe um novo modelo de autenticação, ideal para implementar nas fronteiras das redes de nuvens, e capaz de mediar o acesso de utilizadores aos serviços subscritos enquanto remenda as ameaças de segurança que emancipam da Internet. Este modelo funciona como um ponto central de segurança onde podem ser forçados controlos de segurança de qualquer tipo e a sua maior vantagem provém do facto de recorrer à tecnologia própria da computação na nuvem, nomeadamente virtualização, das quais herda robustez e elasticidade. Um protótipo que mostra como o modelo pode ser implementado é ainda descrito nesta dissertação. Este protótipo usa o cartão de cidadão Português para conseguir autenticação forte e mútua através de certificados de chave pública, de forma simples e transparente para

o utilizador. O protótipo demonstra a funcionalidade do modelo proposto e como pode ser implementado utilizado a tecnologia de computação na nuvem, de forma a esconder a interface interna de ameaças externas. Recomendações para implementar métodos de autenticação são também descritos, fluindo a discussão de métodos clássicos para autenticação para abordagens tendenciais de autenticação.

Palavras-chave

Autenticação, Autenticação por Múltiplos Fatores, Computação na Nuvem, Criptografia de Chave Pública, Interfaces de Gestão, Nuvens Públicas, Segurança

Extended Abstract in Portuguese

Introdução

Este capítulo, escrito na língua Portuguesa, condensa as ideias principais discutidas ao longo do corpo desta dissertação. De seguida, são brevemente delineados o seu enquadramento, a descrição do problema e os objetivos propostos. Posteriormente, as contribuições principais do trabalho representado por esta dissertação são enumeradas e brevemente descritas. Os conceitos mais importantes do modelo de computação na nuvem são então explicados, antes de vários métodos para realizar autenticação serem discutidos. O estudo apresentado na respetiva secção inclui uma revisão da literatura. Na ante-penúltima secção, inúmeros problemas de segurança são apontados como causas para a reputação da segurança da computação na nuvem. Adicionalmente, a segurança de métodos e tecnologias de autenticação também são revistos. Com base neste estudo, um modelo para uma autenticação segura é proposto, juntamente com um protótipo para mostrar a sua aplicabilidade. Recomendações para construir mecanismos seguros de autenticação são também apontados. Finalmente, as conclusões e o trabalho futuro são incluídos na última secção do capítulo.

Neste capítulo e no Resumo anterior os acrónimos estão definidos na língua Portuguesa. Contudo, a forma curta de cada acrónimo utilizado está de acordo com a forma curta do respetivo acrónimo na língua Inglesa, de maneira a manter consistência. Adicionalmente, alguns acrónimos poderão não ter tradução direta (*e.g.*, protocolos), pelo que é mantida a sua designação inglesa e tratados como estrangeirismo.

Enquadramento, Descrição do Problema e Objetivos

No passado, a computação era realizada em grandes unidades de processamento que ocupavam uma sala inteira, requerendo vários técnicos para gestão dos recursos computacionais e de rede. Desde esses tempos, a tecnologia evoluiu a uma grande velocidade. Hoje em dia, a computação na nuvem é debatida simultaneamente pela indústria e pela academia. O primeiro ponto de discussão recai sobre os seus benefícios claros para os clientes e utilizadores de computação na nuvem, que reduzem custos enquanto promovem os seus negócios ou aplicações. Por outro lado, esta tecnologia também é largamente alvo de escrutínio devido aos problemas de segurança que levanta. Um dos principais causadores deste ponto de discussão é a Internet que tem, ao longo dos anos, vindo a ser assolada por um clima de ciberguerra e de cibercriminalidade [SE13], preenchido por várias ameaças. Dado que o modelo de computação na nuvem implica migrar infraestruturas de tecnologias da informação para servidores de terceiros, a Internet tem um papel importante como meio de comunicação. Isto implica que as interfaces de gestão da computação na nuvem fiquem expostas aos perigos que a Internet apresenta, particularmente no que diz respeito à autenticação. O problema é magnificado pela degradação da segurança

associada à utilização de palavras-passe.

Computação na nuvem é ainda sinónimo de problemas relacionados com confiança [Pea13], no sentido em que terceiras entidades ficam responsáveis por dados potencialmente sensíveis e críticos para clientes empresariais. As leis que governam a computação também não são claras [ZZX+10] relativamente a este novo paradigma. Mais do que isso, também não é certo onde os dados de determinados clientes residem na nuvem [XX13], já que a virtualização, aliada à elasticidade e balanceamento automático de recursos permite migrar VMs. Neste contexto é possível enumerar ataques [RTSS09] que exploram o ambiente de virtualização partilhado. As interfaces de gestão da computação na nuvem oferecem funcionalidades de valor crucial. Dado esse facto, é importante garantir a segurança deste ponto específico de acesso aos recursos do cliente garantindo, entre outros requisitos, que a autenticação é forte. Embora reconhecidamente fraca, a combinação de nomes de utilizadores com palavras-passe é uma opção ainda bastante escolhida.

O uso de palavras-passe para efeitos de autenticação é anterior ao uso do computador e foi adotado logo que foi necessário autenticar uma entidade perante uma máquina quer de forma *offline*, quer *online*. Na altura, este método era suficiente e aplicava-se bem, já que o método de entrada era baseado em texto e o poder computacional era pouco. Mas, ao longo dos anos, imensas fugas [Ker13a] de palavras-passe proporcionaram a atacantes métodos [Daz] para construir algoritmos eficientes [KKM+12] para encontrar valores de *hash* correspondentes a determinadas palavras-passe. O *hardware* também tem vindo a ficar cada vez mais poderoso, facilitando ataques de força-bruta e de dicionário [Goo12]. Para ultrapassar estes problemas, a indústria e a academia têm-se focado no desenho e desenvolvimento de mecanismos de autenticação alternativos. Muitos mecanismos são baseados em Autenticação por Múltiplos Fatores (MFA) [GU13, YJIZ12], adicionando camadas de segurança adicionais para além da palavra-passe habitual. Estes podem ser integrados com autenticação por Sistemas de *Login* Único (SSO) [CLJ+11] ou sistemas baseados em risco [TWO+12]. Contudo, todas estas abordagens são vulneráveis aos problemas herdados da Internet, que motivam parcialmente o facto das nuvens públicas concretizarem alvos de ataque apetecíveis.

Esta dissertação apresenta um trabalho que se enquadra nos campos de *segurança na computação em nuvem* e de *autenticação*. O trabalho recai sobre o estudo do modelo de computação na nuvem bem como os problemas que apresenta, focando-se particularmente no estudo da autenticação em interfaces de gestão, revendo entretanto as abordagens existentes para autenticação. O problema deve ser atacado através da criação de um modelo robusto para realizar autenticação, bem como mostrar a sua aplicabilidade através da implementação de um protótipo ao utilizar a própria tecnologias de computação na nuvem. Adicionalmente, deverão ser enumeradas várias sugestões para construir métodos fortes de autenticação. Para atingir os objetivos propostos, o trabalho de mestrado foi estruturado de acordo com as seguintes tarefas:

1. O primeiro passo passa por contextualizar o problema em estudo e preparar o trabalho de investigação restante. Para começar, o paradigma de computação na nuvem deve ser estudado de forma a identificar os seus fundamentos e os conceitos mais importantes para as tarefas restantes.

2. A segunda tarefa consiste em rever publicações académicas e industriais de forma a elaborar uma perspetiva concisa to estado de segurança de computação na nuvem.

3. A terceira tarefa foca-se apenas na autenticação ao começar pela identificação de mecanismos de autenticação disponíveis na literatura, evoluindo para soluções que os implementam na realidade.

4. A quarta tarefa consiste em analisar os mecanismos identificados na tarefa anterior, nomeadamente ao apontar as suas vantagens e desvantagens e, se possível, fornecer exemplos de incidentes reais para corroborar os problemas discutidos. Esta tarefa também deverá distinguir tendências atuais no que toca a mecanismos de autenticação e das tecnologias utilizadas para o efeito, de forma a suportar a próxima fase do programa.

5. A quinta tarefa relaciona-se com a proposta de um modelo para autenticação na interface de gestão de modelos públicos de computação na nuvem, a qual simultaneamente segue as tendências, é retrocompatível com mecanismos atuais, e utiliza tecnologia específica da computação na nuvem. Esta tarefa deverá, adicionalmente, incorporar uma elaboração de recomendações acerca da correta implementação de procedimentos de autenticação, com base naquilo que foi aprendido nas tarefas anteriores.

6. A última tarefa compreende a implementação de um protótipo que adere ao modelo proposto, mostrando a sua aplicação e, potencialmente, destacando as melhores características da proposta. O protótipo deverá ser um exemplo de uma implementação de um mecanismo de autenticação simples, mas ao mesmo tempo seguro.

Contribuições Principais

Esta secção apresenta sintetizadamente as principais contribuições científicas resultantes do trabalho apresentado nesta dissertação. As contribuições principais podem ser sumariadas da seguinte forma:

1. A primeira contribuição incide sobre uma perspetiva completa do estado de segurança da computação na nuvem, a qual foi elaborada durante o programa de mestrado, dentro do âmbito deste trabalho, e que foi mais focada em problemas de segurança relacionados com a autenticação. Este trabalho incluiu rever a literatura, compilar e organizar um número significante de referências interessantes neste campo e propor uma taxonomia para classificar problemas de segurança específicos a ambientes na nuvem. Este trabalho de investigação reverteu para duas publicações: um capítulo aceite para publicação no

livro titulado de *Security, Privacy and Trust in Cloud Systems* [SFG+14], publicado pela Springer; e um artigo [FSG+13b] aceite para publicação numa edição especial da revista *International Journal of Information Security* titulada de *Security in Cloud Computing*, publicado pela Springer, a qual tem um fator de impacto de 0.480 segundo o *Journal Citation Reports* 2012, publicado pela Thomson Reuters. A primeira publicação consiste num estudo da literatura no tópico de computação na nuvem com o foco principal direcionado para a segurança. Para além de rever os conceitos básicos deste paradigma de computação, enumera vários problemas de segurança. A segunda publicação é uma continuação do trabalho da primeira contribuição. Contudo, esta continuação alarga o âmbito de estudo e ainda melhora os conteúdos discutidos nela. São enumerados mais problemas de segurança que afetam os ambientes de computação na nuvem, propondo subsequentemente uma taxonomia para as suas classificações. São debatidas ideias para melhorar sistemas de computação na nuvem ao adotar boas práticas na área e abordagens específicas da tecnologia.

2. A segunda contribuição é a enumeração de mecanismos atuais para autenticação na interface de gestão de ambientes de computação na nuvem, juntamente com os problemas que estas interfaces poderão sofrer, assim atingindo um dos principais objetivos deste programa de mestrado.

3. A terceira contribuição principal é a proposta do modelo para autenticação segura de utilizadores em interfaces de gestão de computação na nuvem, ao tirar partido da tecnologia de computação da nuvem na sua forma de virtualização. O protótipo do modelo resulta também do trabalho inerente a esta contribuição. Inicialmente, são revistos protocolos e mecanismos de autenticação, iterando para as tendências de autenticação consideradas na indústria e na academia. Depois, é incluída uma análise dos problemas de segurança para autenticação. O modelo é então descrito, juntamente com o protótipo que usufrui do cartão de cidadão Português. Este estudo e protótipo irá ser incluído nas atas da 32nd IEEE *International Performance Computing and Communications Conference* (IPCCC) [SFFI13], a qual se vai realizar em San Diego, California, EUA, entre 6 e 8 de dezembro, 2013, e cujas atas irão ser publicadas pela IEEE Computer Society.

Embora surja de trabalho colateral, uma quarta contribuição de trabalho relacionado com esta dissertação consiste no delinear de uma perspetiva sobre o estado atual da cibersegurança. É uma análise compacta deste tópico, realizada durante o primeiro semestre de 2013. Em cada dia, uma lista com notícias interessantes publicadas na comunidade de cibersegurança foi compilada, nomeadamente de temas como *software* malicioso, *spam*, *phishing*, e vulnerabilidades. Este estudo concretizou-se num capítulo [FSG+13a] aceite para publicação no livro designado por *Emerging Trends in Information and Communications Technologies Security*, publicado pela Elsevier (Morgan Kaufman).

Estado da Arte da Computação na Nuvem e da Autenticação

O estado da arte da computação na nuvem e da autenticação é apresentado no capítulo 2. A revisão destes tópicos permite perceber até que ponto a computação na nuvem é importante hoje em dia, e também justificar a necessidade de autenticação forte na Internet de hoje em dia. O capítulo começa por introduzir os conceitos básicos do modelo de computação na nuvem, tentando, sempre que possível, alargar o âmbito das discussões para as temáticas da segurança. Esta abordagem permite retirar os pontos fulcrais do paradigma e vários modelos que o definem, pelo que foi dito que este veio revolucionar o mundo das tecnologias da informação. É, no entanto, ainda relativamente recente. Neste sentido, a academia tem-se focado em tornar este modelo de computação seguro, como mostrado na secção 2.2.4, visto que apresenta um leque amplo de problemas de segurança.

Por outro lado, a autenticação, na sua forma clássica, é baseada em palavras-passe, ao passo de que na *web* são utilizados *cookies* [BH12] para identificar utilizadores previamente autenticados a longo prazo. A complexidade da utilização e gestão das palavras-passe tem aumentado, já que cada utilizador pode estar registado em dezenas de aplicações. Assim, surgiram gestores de palavras-passe [Rei], bem como maneiras de proteger fugas de palavras-passe [JR13] (pois são comuns). Esquemas mais seguros de autenticação são baseados em criptografia, adotando protocolos de chave pública ou de conhecimento zero. A investigação nesta área tem-se ainda focado no conceito de MFA, abraçando várias técnicas de SSO [McA13a], códigos de Resposta Rápida (QR) [CLJ+11], ou mensagens do Serviço de Mensagens Curtas (SMS), sendo que a abordagem particular de Autenticação por Dois Fatores (2FA) tem ganho popularidade [GU13, RSAa]. No último caso, a grande parte dos mecanismos de autenticação utilizam protocolos de Palavra-Passe Única (OTP). A abordagem de autenticação baseada em risco [RQSL12] também tem crescido de forma peculiar. Por outro lado, existem esforços [FID, OAT] para criar protocolos que sejam interoperáveis e universais para transformar a autenticação como ela é percecionada hoje em dia. A meta final é tornar os mecanismos de autenticação mais fortes ao se focarem nos dispositivos e nos utilizadores e ao deixarem as palavras-passe para trás.

Segurança na Computação na Nuvem e na Autenticação

O paradigma de computação na nuvem é vantajoso pelas suas caraterísticas inerentes, mas traz bastantes problemas de segurança associados à sua utilização. Adicionalmente, os mecanismos existentes de autenticação começam a perder a sua força, enquanto que os novos que surgem poderão ser melhorados a nível de segurança. Tais problemas de segurança são enumerados e discutidos no capítulo 3. Em principal destaque está a Internet. Esta rede que interliga outras redes à escala mundial é utilizada para vários fins comerciais, de lazer, de aprendizagem, entre outros. Contudo, uma parte dos seus utilizadores utilizam-na com objetivos maliciosos [PA12]. A partir destes, surge um conjunto variado de ameaças, como *software* malicioso, *spam*, *phishing*, ou ataques de negação de serviço. Os governos têm vindo a participar para

este estado ativamente, contratando cibercriminosos ou formando *hackers* para conduzir ataques contra outros países ou, talvez mais preocupante, levando a cabo programas de vigilância em massa [Sch13a].

Em segundo lugar, a tecnologia de computação na nuvem, como dito na secção anterior, trouxe vários problemas de segurança. Tais problemas de segurança podem ser divididos em virtualização, armazenamento e computação, confiança, legalidade, e cumprimento, bem como *software*, acesso, e rede no âmbito das interfaces de gestão. Ataques [RTSS09, RC11] à virtualização exploram o facto de que máquinas virtuais (VMs) correm lado-a-lado sobre o mesmo *hardware*, enquanto que o armazenamento e a computação em nuvens implica uma multi-localização [ZZX+10], ainda havendo a possibilidade de computação desonesta [XX13]. Uma nuvem pública dita que os dados dos clientes são armazenados em servidores controlados pelos fornecedores de nuvens. Neste sentido, os clientes necessitam de depositar grandes quantidades de confiança [Pea13] neles, bem como nas próprias máquinas. Adicionalmente, as leis estão desatualizadas [ZZX+10] face a este paradigma de computação, enquanto que não é certo o que é que os fornecedores de nuvens podem fazer livremente com os dados dos clientes. As interfaces de gestão de computação na nuvem são construídas com tecnologia *web*, pelo que os problemas [Tro13, JGH09] desta também incorrem nas aplicações. O problema é mais grave para as interfaces permitem a gestão de VMs, já que um adversário com uma única conta comprometida pode causar bastantes danos. Ao contrário das interfaces de gestão, que costumam estar dentro do perímetro de confiança e segurança em redes convencionais, as interfaces de computação na nuvem estão expostas à Internet.

A segurança das palavras-passe tem vindo a degradar-se ao longo dos tempos. Um dos fatores principais para esta queda são a consciencialização que os utilizadores têm acerca das ameaças existentes na Internet. Alguns escolhem palavras-passe com poucos carateres que são fáceis de adivinhar [Tru13], enquanto que outros as deixam em locais explícitos ou as comunicam a outras pessoas [Tow09]. Um outro fator está relacionado com as várias fugas de palavras-passe [Liv13, Ker13a]. Estas fugas ajudam atacantes a juntar dicionários grandes [57u13] e desenhar algoritmos eficientes [KKM+12] para encontrar o valor de *hash* correspondente a determinadas palavras. Isto, aliado com *hardware* de processamento poderoso, facilita ataques de força-bruta e do dicionário [Goo12].

Em relação à segurança das novas abordagens para autenticação, pode ser dito que, apesar de algumas fraquezas que possam apontar, têm vindo a robustecer os mecanismos de autenticação em geral. É preferível ter um segundo fator de autenticação ativo quando o primeiro é comprometido. Os principais pontos de ataque aos OTPs são os servidores [11] que correm os algoritmos e os dispositivos que recebem os códigos via SMS ou os geram via aplicações *offline*. Nestes casos, *software* malicioso e *phishing* são as principais ameaças [Web13]. Adicionalmente, até os códigos QR se mostram vulneráveis a ameaças particulares [KLM+10]. Além disso, as vulnerabi-

lidades [Gre13] dos dispositivos móveis também podem ter impacto no 2FA, quando o objetivo é ultrapassar a autenticação no dispositivo. A motivação principal para o trabalho apresentado nesta dissertação surge dos problemas apontados nesta secção.

Autenticação Segura de Utilizadores em Ambientes na Nuvem

O capítulo 4 detalha um modelo proposto nesta dissertação para autenticação segura em ambientes de computação na nuvem, bem como o protótipo que o implementa. Este capítulo também inclui algumas recomendações. O modelo proposto está esquematizado na figura 4.1. É composto por pelo menos duas VMs ligadas por uma rede privada virtual. Uma das VMs contém a interface de gestão, enquanto que a outra executa funções de *proxy gateway*. Esta arquitetura é inspirada no Whonix [Who], que esconde a máquina interface do exterior, e no trabalho de Salah *et al*. [SACZ+13]. A VM de *proxy gateway* funciona como um mediador dos acessos à interface, reencaminhado o tráfego legítimo convenientemente para ela e bloqueando aquele não desejado. O objetivo é o de minimizar a exposição às ameaças da Internet ao colocar controlos de segurança no *proxy gateway*, virtualizados ou não. Esta abordagem é vantajosa no sentido de utilizar a própria tecnologia de computação na nuvem, sendo que a elasticidade e a rápida instanciação de VMs são úteis para situações atípicas de processamento. Também usufrui do facto de ser possível implementar uma primeira camada de autenticação no *proxy gateway*, deixando um segundo fator para a interface de gestão em si. O modelo é adequado para servir de infraestrutura segura onde podem ser implementados protocolos de autenticação seguros, segundo a arquitetura definida e ao seguir as ideias discutidas.

A figura 4.2 ilustra o funcionamento do protótipo implementado no âmbito do modelo proposto. O protótipo foi construído com duas VMs dispostas como o modelo indica, pelo que apenas tráfego dirigido ao porto 443 (*HyperText Transport Protocol Secure* (HTTPS)) é permitido. Tudo o resto é barrado na *firewall* do Linux, o qual foi instalado em ambas as VMs. O cartão de cidadão Português foi utilizado para autenticação mútua segura utilizando certificados de chave pública. O cartão de cidadão Português incorpora um microprocessador criptográfico e certificados para autenticação e assinaturas digitais, fazendo dele um método seguro de 2FA, para além de ser simples de usar. Para o protótipo funcionar, é necessário configurar o *middleware* do cartão de cidadão no Mozilla Firefox, um *web browser* que permite esta operação. O servidor da interface de gestão foi configurado para levar a cabo autenticação mútua a nível da camada de sessão, estabelecendo uma sessão apenas caso os certificados trocados sejam válidos. As configurações no lado do cliente seriam facilmente ultrapassadas ao seguir uma norma para *smartcards*.

Idealmente, alguns dos problemas apontados anteriormente podem ser ultrapassados ao seguir um conjunto de boas práticas para o processo de autenticação. Os códigos QR fornecem uma experiência agradável ao utilizador e são capazes de codificar material criptográfico durante a

execução de um protocolo, como é o caso de Authentify xFA [Aut]. Porém, o dispositivo móvel utilizado para a leitura dos códigos necessita de estar bem protegido contra *software* malicioso, por exemplo. É útil tornar os dispositivos móveis em autenticadores pessoais, mas as devidas precauções necessitam de ser tomadas. Do ponto de vista empresarial, é difícil concretizar as políticas de segurança nos dispositivos dos seus empregados, pelo que é necessário cooperação entre ambos. Relativamente às comunicações, o HTTPS deverá ser sempre implementado sem construir páginas dinâmicas que buscam outros recursos de forma insegura, sem cifragem das comunicações. Deverão ser seguidas regras rigorosas na construção de aplicações para a nuvem [Mar13], dando ênfase particular à autenticação. Também é importante integrar ativamente os utilizadores na criação de material criptográfico, como as suas chaves criptográficas. Por exemplo, a Amazon *Elastic Compute Cloud* (EC2) gera as chaves na nuvem, enquanto que o serviço MEGA as gera no lado do utilizador. O *software* de 2FA da Google [Gooc], chamado de Google *Authenticator*, pode ser integrado no *kernel* do Linux. Para VMs em nuvens, esta seria uma medida interessante de segurança quando são acedidas remotamente. Sabendo que, apesar das palavras-passe estarem em declínio, o seu legado nos sistemas atuais não é desprezável. Para dificultar os ataques de força-bruta, deverão ser adotados algoritmos que calculem os valores de *hash* lentamente [Hal10], ao invés de alguns que são popularmente utilizados.

Conclusões e Trabalho Futuro

As conclusões e trabalho futuro são discutidas no capítulo 5. O modelo de computação na nuvem nasceu da constante evolução da tecnologia. Este paradigma computacional está a movimentar massas da indústria e da academia, visto trazer bastantes benefícios, mas também problemas de segurança. Baixar custos enquanto aumenta produtividade é vantajoso para o cliente. Contudo, as interfaces de gestão da computação na nuvem sofrem de vários problemas de segurança inerentes à própria tecnologia e também provenientes do exterior, a Internet. A virtualização é um elemento chave na nuvem, mas num ambiente partilhado aumenta o risco de ataques de VM para VM. Também o cumprimento legal num ambiente de computação na nuvem não é claro, visto que as leis não se aplicam a este novo modelo. Por outro lado, a Internet é hoje em dia utilizada tanto para o bem como para o mal. Assim sendo, as interfaces estão sob o risco constante de ataque, o que torna a autenticação nestas um procedimento crucial.

Atualmente, os mecanismos de autenticação atravessam uma fase de transição. As abordagens clássicas estão a sucumbir perante técnicas modernas, ficando para trás, enquanto que métodos mais fortes e alternativos estão em desenvolvimento. As palavras-passe são facilmente apanhadas por *phishing* ou *software* malicioso. A tecnologia móvel está cada vez mais a ter um papel preponderante no dia-a-dia dos utilizadores. Esta dependência é aproveitada para construir mecanismos de segurança mais seguros ao seguir a abordagem do 2FA baseada em *tokens* de *software* ou *hardware*, quer seja via mensagens de SMS ou via códigos QR, por exemplo. As tendências são claras, pelo que é necessário adaptar os métodos atuais de autenticação para

as redes atuais, tendo em conta o estado atual da Internet. Para minimizar o potencial impacto de ataques direcionados às interfaces de gestão, esta dissertação apresentou um modelo robusto para autenticação segura de utilizadores. O modelo é baseado na própria tecnologia de computação na nuvem, pelo que utiliza duas VMs em que uma faz de *proxy gateway* para a outra, que contém a interface de gestão. O papel do *proxy gateway* é o de mediar e controlar os acessos à interface contra ameaças externas, entretanto aproveitando os benefícios da tecnologia de computação na nuvem. Então, o modelo vem, também, aliviar o problema, dando a clientes uma forte motivação para adotarem soluções de computação na nuvem enquanto que a segurança é logo reforçada em primeiro lugar. Para mostrar a aplicabilidade do modelo, foi implementado um protótipo que utiliza o cartão de cidadão Português para levar a cabo autenticação forte e mútua ao utilizar criptografia de chave pública e certificados. O *proxy gateway* centraliza funções de segurança, pelo que é necessário averiguar a sua eficiência sobre condições atípicas de funcionamento (*e.g.*, sobre um ataque). Como é que os vários controlos de segurança podem ser combinados para criar determinados perfis de segurança e como cada um melhor responde para determinados casos é também deixado como trabalho futuro.

Com isto, torna-se claro que é necessário que a segurança seja priorizada, e em particular na autenticação já que é a primeira barreira de segurança para determinados recursos. Neste sentido, e necessário adotar medidas mais fortes baseadas em Criptografia sobre Curvas Elíticas (ECC) e ao adotar chaves criptográficas de maior dimensão [Sch13b], no caso de esquemas baseados em criptografia. Eventualmente, a segurança deverá ser encarada de forma mais bem definida em serviços baseados em computação na nuvem, enquanto que a autenticação forte será um peça crítica destes sistemas.

Abstract

For a handful of years, *cloud computing* has been a hot catchphrase. The industry has massively adopted it and the academia is focusing on improving the technology, which has been evolving at a quick pace. The cloud computing paradigm consists in adopting solutions provisioned by some cloud providers that are hosted on data centers. Customers are therefore tied to those third-party entities, since they becomes involved in their businesses for being responsible for the Information Technologies (IT) infrastructures outsourced to the clouds. This implies that customers have to totally or partially migrate their on-premises infrastructures to off-premises clouds, including, but not limited to, email, web applications, storage databases, and even complete servers that become wrapped in services accessed via the Internet. Clouds deliver scalable and elastic networking, storage, and processing capabilities in an on-demand and self--provisioned manner adopting the pay-as-you-go business model. This benefits the customers significantly, allowing them to promote their businesses without worrying about inherent IT infrastructures.

The services supplied by clouds are basically encapsulated by one of the three main service delivery models: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). They are the building blocks for unfolding Anything-as-a-Service (XaaS) solutions specifically customized to customer requirements. IaaS mixes novel virtualization techniques with current technologies that allow to run Operating Systems (OSes) or even build entire virtual data centers. PaaS allows to develop applications in a consistent manner via the cloud platforms to run remotely, while SaaS enables enjoying pre-built software with little control over the application flow. These models can run on public or private clouds, or on a hybrid version of the two. Adopting the public model means accessing the subscribed services through the Internet from anywhere in the globe. IaaS clouds usually have some management interface to control Virtual Machines (VMs) and to arrange a virtual data center, while VMs are accessed via standard remote connection protocols. The authentication to those interface is, therefore, of utmost importance, mostly because they are exposed to Internet dangers, contrarily to traditional management tools that are deeply within the trusted perimeter of a company on conventional networks. This dissertation first identifies such problems by reviewing authentication approaches and pointing out their advantages and weaknesses. For example, a single compromised cloud account constitutes an inherently more dangerous threat when compared to traditional website accounts, because an attacker gains control over VMs and potentially over security-related configurations as well. This can result in data and money losses for both customers and providers, since the malicious attacker can terminate VM instances running crucial business applications.

The problem also resides on the fact that the security of authentication by means of a static

password has been decreasing over time, and it is now a method considered unsafe. Data breaches in the past have provided key understanding over user password habits, allowing crackers to build huge password lists and devise efficient cracking algorithms. Current threats that are inherent to the Internet technologies aggravate this problem further. The cyberspace is increasingly getting more violent with cyberwarfare, cybercriminality, and mass surveillance affecting everyone. For instance, malware writers are focusing on mobile platforms since that is a field where technology is rapidly evolving, sometimes in an uncontrolled manner, and that users are widely adopting. As a reaction to some of those problems, both the industry and the academia have turned their attention to alternative authentication schemes based on Single Sign-On (SSO) and Multi-Factor Authentication (MFA). SSO is used to alleviate the burden associated with the management of multiple credentials for cloud applications, while MFA adds authentication layers for additional security, in exchange for a drop in usability and an increase to the cost. Such new approaches implement authentication schemes resorting to both public-key and symmetric cryptography mechanisms and to new technologies, namely associated to mobile. Several contributions on this research line can be found in the recent literature. Nonetheless, they still do not pay particular attention to the underlying infrastructure and threats.

The scope of this dissertation is confined to the topics of cloud computing and authentication, studying how that new computing model works, with particular emphasis on the security, and reviewing authentication methods orthogonally to cloud computing and to the security perspective. Because cloud management interfaces are inherently more riskier for being exposed in the Internet and starting from the previously mentioned study, this dissertation proposes a new security model, suitable to deploy on the frontend edges of cloud networks, mediating the access of users to the subscribed services, while patching security threats inherited from the Internet. It is envisaged as a centralized point for enforcing several types of security controls and its main advantage derives from the fact that it resorts to the usage of technology specific to cloud computing, namely virtualization, from which it inherits robustness and elasticity. A prototype showing how the model can be deployed is also described in this dissertation. It uses the Portuguese identity card to achieve strong and mutual authentication by means of public-key certificates, in a way that is simple and transparent. to the user. The prototype demonstrates the functionality of the proposed model and how it can be deployed using cloud computing technology, so as to conceal the internal interface from outside threats. Recommendations for implementing authentication methods are also described, flowing the discussion from classical authentication to trendy authentication approaches.

Keywords

Authentication, Cloud Computing, Management Interfaces, Multi-Factor Authentication, Public Clouds, Public-Key Cryptography, Security

Contents

List of Figures

List of Tables

Acronyms

2FA	Two-Factor Authentication
AD	Active Directory
API	Application Programming Interface
APT	Advanced Persistent Threat
ARP	Address Resolution Protocol
ATM	Automated Teller Machine
AWS	Amazon Web Services
BBAuth	Browser-Based Authentication
BYOD	Bring Your Own Device
CA	Certificate Authority
CaaS	Cybercrime-as-a-Service
CnC	Command-and-Control
CPU	Central Processing Unit
DaaS	Data-as-a-Service
DHCP	Dynamic Host Configuration Protocol
DLL	Dynamic-Link Library
DLP	Discrete Logarithm Problem
DDoS	Distributed Denial of Service
DDoSaaS	Distributed Denial of Service-as-a-Service
DoS	Denial of Service
DSA	Data Signature Algorithm
EC2	Elastic Compute Cloud
ECC	Elliptic Curve Cryptography
FIDO	Fast IDentity Online
GAE	Google App Engine
GPU	Graphics Processing Unit
HMAC	Hash-based Message Authentication Code
HOTP	HMAC-based One-Time Password
HTTP	HyperText Transport Protocol
HTTPS	HyperText Transport Protocol Secure
IaaS	Infrastructure-as-a-Service
IDE	Integrated Development Environment
IdP	Identity Provider
IDS	Intrusion Detection System
IETF	Internet Engineering Task Force
IPv4	Internet Protocol version 4

IPv6	Internet Protocol version 6
ISP	Internet Service Provider
IT	Information Technologies
LDAP	Lightweight Directory Access Protocol
MAC	Message Authentication Code
MD5	Message Digest 5
MFA	Multi-Factor Authentication
MitB	Man-in-the-Browser
MitM	Man-in-the-Middle
NAT	Network Address Translation
NFC	Near Field Communication
NIST	National Institute of Standards and Technology
NOC	Network Operations Center
NSA	National Security Agency
OASIS	Organization for the Advancement of Structured Information Standards
OATH	Initiative for Open AuTHentication
OCSP	Online Certificate Status Protocol
OCRA	OATH Challenge-Response Algorithm
OS	Operating System
OSTP	Online Security Transaction Protocol
OTP	One-Time Password
P2P	Peer-to-Peer
PaaS	Platform-as-a-Service
PAM	Pluggable Authentication Module
PEM	Privacy Enhanced Mail
PGP	Pretty Good Privacy
PHP	Hypertext Preprocessor
PIN	Personal Identification Number
PKCS	Public-Key Cryptography Standard
PKI	Public-Key Infrastructure
PoC	Proof-of-Concept
PRNG	Pseudo-Random Number Generator
PUE	Power Usage Effectiveness
QR	Quick Response
RDP	Remote Desktop Protocol
RFC	Request for Comments
RFID	Radio-Frequency IDentification
RSA	Rivest, Shamir, Adleman
SaaS	Software-as-a-Service

SAML	Security Assertion Markup Language
S_{ec}aaS	Security-as-a-Service
SDK	Software Development Kit
SDLC	Software Development Life Cycle
SHA-1	Secure Hash Algorithm-1
SIEM	Security Information and Event Management
SLA	Service Level Agreement
SMS	Short Message Service
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
SOC	Security Operations Center
SQLi	Structured Query Language Injection
SSH	Secure Shell
SSO	Single Sign-On
SVA	Security Virtual Appliance
TAN	Transaction Authentication Number
TLS	Transport Layer Security
TOC	Technology Operations Center
Tor	The Onion Routing
TOTP	Time-based One-Time Password
TPM	Trusted Platform Module
UAC	User Account Control
URL	Uniform Resource Locator
USB	Universal Serial Bus
UTF-8	UCS Transformation Format–8
VCPU	Virtual Central Processing Unit
VDI	Virtual Desktop Infrastructure
VM	Virtual Machine
VMM	Virtual Machine Monitor
VNIC	Virtual Network Interface Card
VPC	Virtual Private Cloud
VPN	Virtual Private Network
VPS	Virtual Private Server
WAF	Web Application Firewall
XaaS	Anything-as-a-Service
XML	eXtensible Markup Language
XSS	Cross-Site Scripting
ZKP	Zero-Knowledge Protocol

Abbreviations

Please consider the following abbreviations with the respective meaning when later invoked in the text:

e.g.	originates from the Latin expression *exempli gratia* which means “for example.”
i.e.	originates form the Latin expression *id est* which means “that is” or “in other words.”

Chapter 1

Introduction

This dissertation elaborates on the subject of securely authenticating users on cloud management interfaces with respect to emerging trends and issues in this area. The focus and scope of the dissertation are next described, along with the problem statement and objectives, the adopted approach for solving the problem, the main contributions, and the dissertation overview.

1.1 Focus and Scope

In the past, computing was mainly carried out at large time-shared mainframes held in big rooms which would output (relatively) little processing power. Computer terminals would dial into local machines or would conduct long-distance calls to countries abroad, to some university or government machines with simple username and password authentication on an era devoid of security awareness. Taxes would accordingly apply similarly to what happens today in telecommunications. Distinct rates exist among different operators, but national calls are typically cheaper than the international ones. As an overlay network over the existing telephony lines, the Internet emerged to connect local networks around the globe via a public infrastructure. Nowadays, there is an ongoing smooth transition to fiber optics technology, which offers greater speed and lower latency, thereby providing higher reliability for increasingly demanding Internet communications. Because of the usefulness and commodity of the Internet in supporting businesses or leisure activities, some exploit it for their own profit. The virtual underground is a dark place, from which many security threats emerge from. A cyberwar setting is in place, ranging from hacktivism, passing by nation-states conflicts, to cybercriminality. As such, novel computing technologies, relying on the Internet, become inherently less safer. This is the case of the *cloud computing* technology.

Cloud computing shifts the perception in terms of infrastructures and services by detaching cloud customers from hardware needs. To enterprises, it represents the possibility to migrate entire Information Technologies (IT) infrastructures to some outsourced *cloud*, hosted on a *data center* and managed by a given *cloud provider*. That includes migrating email, web applications, storage databases, and even complete servers to wrapped services. This is done autonomously and on-demand, in a self-provisioned fashion, where the customer chooses what best sees fit, for which a pay-as-you-go business model is incurred. Clouds mainly offer highly scalable and elastic networking, storage, and processing capabilities. Customers access their

services via the Internet in the case of the public cloud deployment model. Whereas public clouds are outside the trusted border, a private cloud can be established within the trusted perimeter, the internal corporate network. A hybrid cloud mixes the two for higher security and trust. Services can be categorized into one of the three main cloud service delivery models: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). The SaaS model offers pre-built software instances while the PaaS model offers platforms for customers to build cloud applications. The IaaS model is perhaps the most revolutionary one. It leverages the virtualization technology for replacing physical servers by umpteen Virtual Machine (VM) instances. IaaS clouds can house particular VMs or large virtual data centers composed of many VMs, virtually connected to simulate the physical analogue. VMs can encapsulate entire Operating Systems (OSes) by creating isolated memory sandboxes and by simulating underlying physical resources provisioned by Virtual Machine Monitors (VMMs), which are programs in charge of managing VMs and related tasks. These models form the means for clouds to deliver Anything-as-a-Service (XaaS), ranging from large resources to granular requirements, which makes of cloud computing a desirable solution. By abstracting hardware needs, customers can thus increase their productivity and focus on promoting their businesses.

The benefits of cloud computing are clear to major players and can be concluded from its rapid growth in the last handful of years. It is a general opinion that cloud computing is inevitably a technology that will remain around for the near future. Clouds unfold a whole new panorama of computing possibilities, but their security state is in a gray area, due to several issues. Because it is a relatively new paradigm, involving novel technologies, it has not yet matured sufficiently to allow a seamless and smooth transition to this paradigm. This subject is highly debated throughout the industry, the academia and even the media. From the customer viewpoint, data is simply stored on the cloud, without knowing its true physical location on a multi-tenant shared infrastructure. Other customers access the cloud as well, but there may be no assurance regarding their legitimacy nor their goals for using the cloud. Malicious actors can disguise as legitimate customers and roam the cloud *freely* (to some extent), potentially peaking to other customers data by exploiting some vulnerability. Cloud computing has introduced uncharted security issues. For example, the virtualization attack surface is wide, depicting attacks of VM escape or VM-to-VM like side- and covert-channels.

On IaaS cloud environments, each customer has access to a user-friendly interface, whose purpose is to manage all tasks related with the virtualized infrastructure, including instantiating, stopping, terminating, editing, cloning, snapshotting and restoring VM instances. In addition, it may be possible to configure security policies, firewall rules, and other features of the same nature, if the cloud provide such features. In this setting, these interfaces centralize the management of an arbitrary number of VMs while containing important information regarding the arrangement of a virtual data center, the IT infrastructure of a customer. Worse, the interfaces

might depict security information like the names of private keys used for accessing VMs via Secure Shell (SSH). This may aid an attacker to conduct additional, more targeted reconnaissance on public cloud deployments. Thus, a single compromised account can be highly problematic, passing the control of all VMs to an attacker. On traditional production networks, only a handful of administrators have access to critical management interfaces that are deeply within the network behind several security controls and may be in particular Internet Protocol (IP) zoning areas, different from the frontend perimeter. This is clearly neglected on the public cloud model. The services and the interfaces themselves are directly accessible from and exposed to the Internet, which is used as the communications medium. As such, clouds inherit several issues specific to the Internet, namely code injection, phishing, malware, and bandwidth starvation. In addition, the growth of mobile devices has turned both enterprise networks and the Internet into more dynamic virtual places, wherein many kinds of traffic originate from countless distinct devices. Because the interfaces are developed using standard web technology, popular attack vectors like injection apply as well. Regarding authentication, standard username and password combinations are still used nowadays on such interfaces. Hence, cloud management interfaces comprise attractive attack points for malefactors.

The scope of this dissertation is restricted to the fields of *cloud computing* and *security*, and particularly on the *authentication* field. More specifically, it is focused on the security issues that may affect the management interface of a cloud environment and on the authentication procedures that should be implemented to duly protect that interface. The *de facto* standard for classifying works of computer science is the ACM Computing Classification System. The categories and descriptors of the 1998 version of the classification system for the work discussed in this dissertation are C.2 [**COMPUTER-COMMUNICATION NETWORKS**]: Security and protection and K.6.5 [**MANAGEMENT OF COMPUTING AND INFORMATION SYSTEMS**]: Security and Protection–*Authentication*.

1.2 Problem Statement and Objectives

The problem addressed in this dissertation is the (probably lack of) security in the authentication procedures of cloud interfaces used for managing the customer service or services, under a public cloud deployment model, where the issues derived from the Internet also apply. The problem can be further detailed by referring that, for decades, the authentication in computer systems has been performed using combinations of usernames and passwords, which are still used nowadays, in spite of several warnings concerning the danger of doing so along the years. Anticipating a possible crisis in this area, new methods have been proposed throughout the time, though most of them have no practical implementations. Focus is being put by the industry and by the academia on devising stronger methods that deliver a well-known set of security properties that are immune to various attacks. In addition, the focus also falls in the usability and efficiency of the methods, striving to meet a seamless authentication procedure on a few

steps while performing quick calculations. *Per se*, and despite no compromise situation of cloud management interfaces has yet been reported, there is the need to harmonize the plethora of schemes proposed in the broad literature into a single robust and solid model that admits the applicability of current authentication schemes. For various reasons, cloud management interfaces are critical points for the businesses of customers, and thus they comprise attractive attack points. Ideally, such a model would also combat the threats emerging from the Internet in order to make clouds a safer computing place.

The main objective of this dissertation is to first survey the related areas and identify the authentication mechanisms implemented and deployed in practice, as well as the issues that management interfaces may be subjected to. Another main objective is to point out the weaknesses of each mechanism, possibly by pointing out real incidents exploring them. Starting from that, a model for authenticating users on public cloud management interfaces should be proposed. The proposed model should take into consideration current authentication schemes. It should also leverage the cloud computing technology itself. The applicability of the model should be demonstrated by means of the implementation of a prototype. Suggestions for the implementation of the model should be handed out in the light of current technologies and trends as well.

1.3 Adopted Approach for Solving the Problem

To achieve the objectives described in the previous section, this masters program was divided into the following tasks:

1. The first step concerns the contextualization with the problem at hands and preparing to the remaining research work. To start with, the cloud computing paradigm should be studied so as to identify its building blocks and the most important concepts for the remaining tasks.

2. The second task consists of surveying the academic and industry publications to elaborate a concise perspective over the security state of cloud computing.

3. The third task is to focus on authentication alone, starting from the identification of available mechanisms for achieving such purpose in the literature and evolving to the solutions that are actually and currently implemented in practice.

4. The fourth task consists in analyzing the mechanisms identified in the previous task, namely by pointing out their strengths and weaknesses, if possible by providing examples of real incidents corroborating such statements. This task should also comprise the identification of current trends in terms of the adoption of authentication mechanisms and of the utilized technologies, so as support the next phase of the program.

5. The fifth task concerns the proposal of a model for authentication in the management interface of public cloud environments, which simultaneously follows the trends, is backward compatible with current mechanisms, and uses cloud specific technology. This task should additionally incorporate the elaboration of recommendations regarding the correct implementation of authentication procedures, in the light of what was learned during this and previous tasks.

6. The last task comprises the implementation of a prototype adhering to the proposed model, showing its feasibility and potentially highlighting the best characteristics of the proposal. The prototype should be an example of a secure but simple to use and implement authentication mechanism.

1.4 Main Contributions

This section briefly describes the scientific contributions resulting from the research work represented by this dissertation. The main contributions may be structured as follows:

1. The first main contribution comprises a complete perspective over the security state of cloud computing, which was elaborated during the masters program within the scope of this work, and that it was more focused on authentication related issues. This work included surveying the literature, compiling and organizing a significant amount of references on the fields of interest and proposing a taxonomy[1] for classifying security issues specific to cloud environments. This research work was on the basis of two publications: a book chapter accepted for publication in the book entitled *Security, Privacy and Trust in Cloud Systems* [SFG+14], published by Springer; and an article accepted for publication in a special issue of the *International Journal of Information Security* entitled *Security in Cloud Computing* [FSG+13b], published by Springer, which has an impact factor of 0.480 according to the Thomson Reuters Journal Citation Reports 2012.

   The first publication introduces the topic of cloud computing, explaining the basic concepts while stressing out security properties of cloud systems, to then present a review of security issues spanning several topics. The second publication is a follow-up work of the previous one, further extending the survey with more materials into a more complete and comprehensive study of cloud security issues in cloud environments. This article proposes a taxonomy to classify security issues, and finally a brainstorm of open challenges and recommendations is included.

2. The second main contribution is the enumeration of the currently available mechanisms for authentication in the management interface of cloud environments along with the issues such interfaces may face, addressing one of the main objectives of this masters program.

[1] This taxonomy is only included in the referred journal article.

3. The third main contribution is the proposal of the model for secure user authentication in cloud computing management interfaces by leveraging IaaS cloud technology, together with a prototype. The second and third main contributions appear in the proceedings [SFFI13] of the 32nd IEEE International Performance Computing and Communications Conference (IPCCC) that will be held in San Diego, California, USA, between the 6th and the 8th of December, 2013, and whose proceedings will be published by the IEEE Computer Society.

A fourth, collateral, contribution to this work comprised a compact analysis of the current state in cybersecurity. The analysis is the result of an effort performed on a daily basis throughout the first half of 2013. In each day, the infosec community was scanned for compelling feeds on the subject, giving priority to topics like malware, spam, phishing and vulnerabilities. The resulting analysis first introduces the topic of cybersecurity and delimits its scope, but it then gives a quick perspective of the cybercrime and cyberwarfare panorama by discussing trends found in those feeds. This part of the work was the subject of a book chapter accepted for publication in the book named *Emerging Trends in Information and Communications Technologies Security* [FSG+13a], published by Elsevier (Morgan Kaufman). Parts of this work are on the basis of several sentences and sections of this dissertation.

1.5 Dissertation Overview

This dissertation is organized in five main chapters. The **body** of the dissertation contains three chapters, preceded and succeeded by the Introduction and the Conclusions and Future Work chapters, respectively. Each main chapter of this dissertation can be summarized as follows:

- **Chapter 1** introduces the subject of this dissertation by first explaining its focus and scope. The problem statement and objectives are included afterwards, followed by the adopted approach for solving the problem. The main contributions of the work depicted in this dissertation are then enumerated, while the dissertation overview constitutes the final part of the chapter.

- **Chapter 2** starts off by explaining the basics of the cloud computing paradigm. The descriptions of various authentication methods come in second. Along the chapter, the ideas being discussed emphasize security of clouds and of authentication, by reviewing the literature on the several subjects.

- **Chapter 3** outlines the most prominent security issues impacting cloud computing, enumerating along the way what kind of outcome may arise from potential attacks. In then converges to authentication, explaining thoroughly how authentication is endangered on cloud computing management interfaces. This chapter also describes several real world

security incidents related with currently employed authentication schemes, as a means to inherently introduce the mistakes that should not be made in the future.

- **Chapter 4** details the model for secure authentication in cloud environments proposed in this dissertation. The goals of the model, along with the assumptions embraced for devising it, are included in the beginning of the chapter. The core of the model is described and a security analysis is included afterwards. Then, a prototype utilizing the model is presented while identifying the technologies used for constructing it. All necessary configurations for the prototype are represented in the chapter.
- **Chapter 5** concludes this dissertation by first commenting on what may be expected in terms of authentication schemes for clouds in the next few years, as well as the implications to computing in general. It finally presents the main conclusions of this dissertation, together with directions for future work.

Notice that the long form of an acronym is repeated in the first three chapters, namely Resumo, Extended Abstract in Portuguese, and Abstract, and then once more in the Introduction and subsequent chapters, so as to keep consistency along the dissertation.

Chapter 2

State-of-the-Art on Cloud Computing and Authentication

This chapter elaborates on the cloud computing paradigm, introducing its building blocks and discussing the several subjects from the security point of view, when applicable. It also introduces the concept of authentication and reviews the state-of-the-art in terms of authentication procedures, analyzing contributions from both the industry and academy.

2.1 Introduction

Cloud computing has really emerged in the industry in the last part of 2007 and beginning of 2008. Actually, Weiss [Wei07] might have published one of the first articles on cloud computing in December of 2007. Names as the International Business Machines (IBM), Google, Amazon and Microsoft are mentioned as industry pioneers of the buzz created around the latest catchphrase around, at the time: *cloud computing*. Since then, it has evolved to become an unavoidable force for computation as a response to the emerging waves of data, being widely debated throughout the IT world as one leap towards the long-envisioned era of utility computing. Various conferences address this topic alone, such as the EMC World, which is lead by the EMC Corporation, a big player in the field. Particularly, cloud computing is broadly discussed for its puzzling security state, which will be addressed in this dissertation also. In order to support such discussion, Section 2.2 first explains the basic concepts of this computing model.

When computer systems started to appear here and there in the fifties and sixties, the network connectivity between them was in early stages. The major concern was to make local machines work properly without ever being connected to the outside. However, that way of thinking and operating has changed. Today, connectivity is everything for carrying out daily activities on smartphones, desktop computers, or other mobile devices, while relying on the Internet for communications. This increase in connectivity also made authentication a top priority, comprising one of the initial access security measures between a person or entity and a resource. As such, the field of authentication has always deserved especial attention. Following this line of thought, Section 2.3 discusses computer-based classical authentication approaches while Section 2.4 sheds light over new trends.

2.2 Cloud Computing Concepts

The cloud computing technology is revolutionizing mature and well-regarded IT perspectives by offering outsourced and automated on-demand services, wrapping functionalities that may span the entire cloud stack. Outsourced clouds deliver remote pools of resources for storage, processing, or networking purposes. This includes replacing on-premises IT infrastructures, but also implies depositing great amounts of trust on the cloud provider, for it holds sensitive data on some data center. Each Central Processing Unit (CPU), memory, server, and anything else connected to the cloud is seen as just another resource, instead of external hardware that requires some middleware to communicate with. Cloud OSes seamless integrate every distributed resource into a solid computing cloud. Customers subscribe and use services according to a pay-per-use business model, meaning that they only pay for what they use (*e.g.*, computing cycles and network bandwidth), and thus costs are reduced since one does not spend on the installation, management, and upgrading of on-premises platforms.

The National Institute of Standards and Technology (NIST) released in 2011 a formal document defining what it is to compute in the clouds, since its definition was in a gray area at the time, and opinions were diverging. The document [NIS11] was well accepted, and characterized cloud computing according to five essential characteristics, three service delivery models, and four cloud deployment models. The first essential characteristic is *on-demand self-service*, which means that customers can unilaterally request more resources automatically without requiring human interaction. The second is *broad network access* that imply that services can be accessed ubiquitously over the network by thin or thick[1] client platforms. *Resource pooling* refers to the distributed, dynamic, scalable, and multi-tenant environment supporting clouds, by either agglomerating physical and virtual resources. The fourth characteristic is *rapid elasticity*, and it refers to the fact that resources can be allocated and reallocated rapidly and seamlessly, appearing as unlimited to the customer. Finally, *measured service* points out the pay-per-use business model, meaning that each service is measured accordingly. The NIST further distinguishes the service delivery models as SaaS, PaaS, and IaaS, sorted upwardly, while the deployment models dwell into public, private, hybrid, and community. All these are described in the next subsections while giving attention to some security aspects of each one.

Since clouds are mostly perceived as being located within some data center, and since highly sensitive customer data can be on those clouds, data centers should uphold physical security as well, beyond the logical security counterpart. As such, this section includes a discussion of the concepts and security of data center facilities also. In addition, a perspective over

[1] In the client-server architecture, clients refer to the software used to access server applications. Thin clients like smartphone applications or terminals are designed to be especially small so that the bulk of processing occurs on the server. On the other hand, thick clients process the majority of the data offline and only require intermittent communications with the server. Thick clients like browsers are inherently less secure because of local malware. Sometimes, they are also referred to as fat clients.

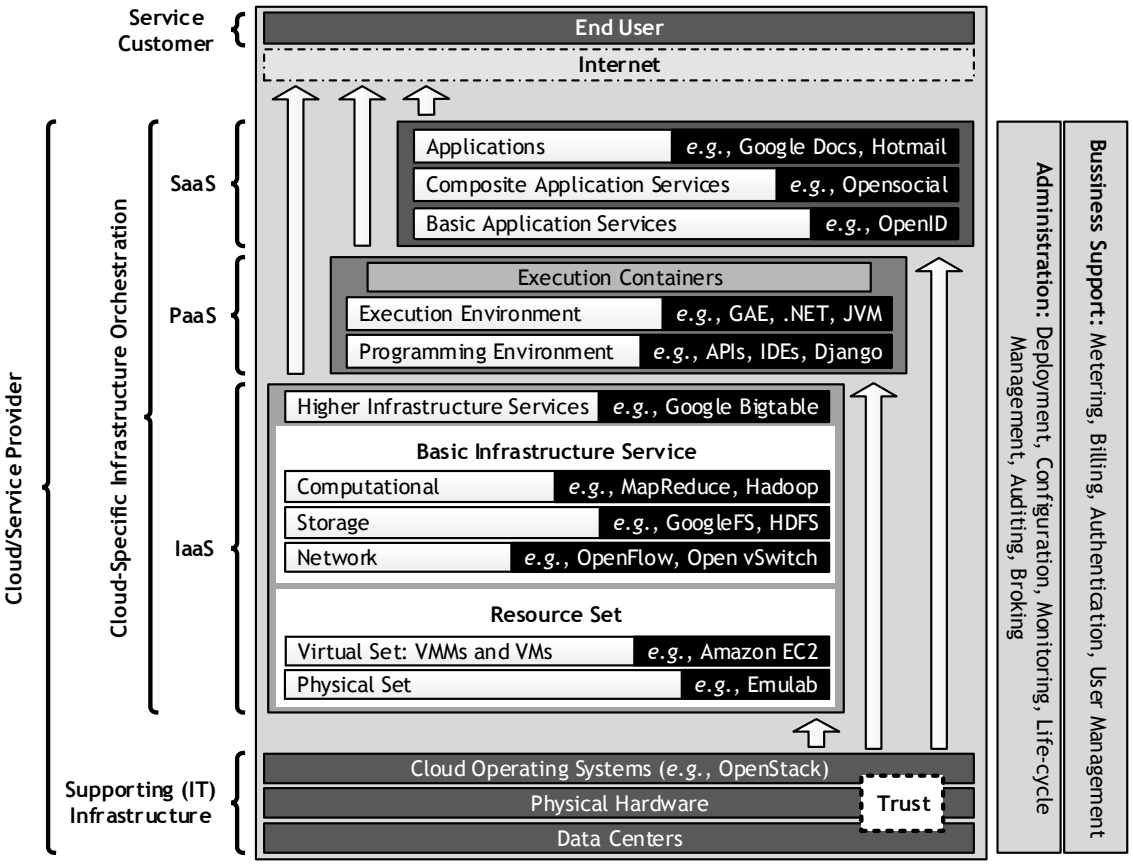

Figure 2.1: Cloud service delivery models depicted with underlying and overlying layers, as well as with the vertical components. This figure is an adaptation of Figure 2 of the second scientific publication [FSG+13b] of this dissertation.

the contributions on the period comprised between the years 2008 and 2012 is given in the end as well. The following discussions are, in part, based on the introductory contents of the first [SFG+14] and second [FSG+13b] scientific publications of the work performed during this masters program.

2.2.1 Cloud Service Delivery Models

It was said above that cloud computing elaborates in three service models: SaaS, PaaS, and IaaS. These make up the core of the cloud stack, or the cloud-specific infrastructure orchestration, as illustrated in Figure 2.1. The figure provides a possible structure of how clouds operate according to the underlying and overlying layers, as well as the vertical interactions between them, coupled with specific business support and administration. A critical aspect of cloud businesses is trust. Trust spans throughout the entire cloud operation, regardless of the cloud service delivery model, being its point of origin the bare-metal itself. Trust then expands from the data centers, on which data resides, to servers, to virtualization technologies, and to everything else that the cloud or the service provider owns and manages entirely.

SaaS, PaaS, and IaaS provide the foundations for clouds to offer XaaS, where the *X* is a wild-

card that is replaced by an acronym for a service. XaaS emphasizes that clouds can be set up to deliver virtually any kind of service, ranging from fine-grained services to large solutions, assuming such services can correspond to the cloud operating and business models. For example, Data-as-a-Service (DaaS) [VPT+12] models specify structured Application Programming Interfaces (APIs) for accessing cloud data sets on-demand, while Security-as-a-Service (S_{ec}aaS) [AALX+12] offers unified threat management for securing cloud services. The security analysis of each of these service models naturally depends on how each is implemented. Nevertheless, SaaS, PaaS, and IaaS can be summarized as follows.

2.2.1.1 Software-as-a-Service

The top model, SaaS, pre-packages specific software that is typically demanding and standard for enterprises. For example, some require ticketing platforms or management applications for the finances or human resources departments. Traditional software may be expensive due to installation and management costs, and may most likely to require a periodic license. In SaaS cloud computing, a thick client, most commonly a browser, is all that is required to access such measured applications, like the ones offered by SmartCloudPT [Pora], a Portuguese SaaS and IaaS cloud provider. Specific APIs may be required and developed so as to support thin clients. SaaS applications do not need to be thoroughly configured, and they work almost out-of-the-box, wherein management responsibilities are left to the cloud provider. The most likely threats are malware using Man-in-the-Browser (MitB) and Man-in-the-Middle (MitM) attack models.

2.2.1.2 Platform-as-a-Service

The middleware model, PaaS, allows to provide customers with frameworks capable of developing and deploying cloud applications. Essentially, PaaS cloud providers offer Software Development Kits (SDKs) and Integrated Development Environments (IDEs) for writing code specifically to run on multi-tenant and shared environments. Each user, or tenant, runs its own applications next to other applications on which some libraries or runtime components might be shared amongst all of them. Applications are encapsulated by containers that are supposed to be consistent with a well defined isolation limit so as to no interfere with threads, processes, or memory belonging to other applications [RMVC+12]. Nonetheless, some clouds might provide pre-packed disk images with some specific software stack defined by the customer. Since customers develop their own cloud applications, they should urge to deploy security features while having in mind the underlying shared framework. In contrast, the cloud provider is responsible for the maintenance and security of the developing frameworks. The Google App Engine (GAE) [Goob] is a well-regarded PaaS provider that allows developing applications in

Java, Python, and Go.

2.2.1.3 Infrastructure-as-a-Service

The bottom model, IaaS, is perhaps the main novelty cloud computing presents to the IT world. Virtualization is the means to emulate hardware and encapsulate OSes with the so-called VMs, leading OSes into thinking they are installed and running over the hardware, but instead are on top of a virtualization layer. This virtualization layer is provided by VMMs, or hypervisors, which are in charge of instancing, destroying, snapshotting[2], restoring, pausing, and replaying VMs with simple commands. VMs can be aggregated in particular virtual network zones, imitating the traditional IP zoning done in enterprise networks.

Hypervisors can provide an arbitrary number of Virtual Central Processing Units (VCPUs), Virtual Network Interface Cards (VNICs), and other emulated hardware to VMs. Hypervisors are typically large complex softwares that can be installed over typical OSes in a *hosted* manner, or can be installed directly in the bare-metal hardware in a *native* manner. As such, hypervisors convert each VM call to the host OS or to the hardware, scheduling VCPUs and other VMs requests accordingly. Such features allow to build entire virtual data centers, which can be seen as mirror views of the real, physical network and IT counterparts. VMs can be rapidly created with little overhead while assigning few resources to that task. Moreover, they are easily configured and can be bounced from one machine to another, migrating contents seamlessly without losing data. Popular paid hypervisors include Parallels Virtuozzo [Par13] and VMware Fusion [VMw13], while free hypervisors include the open-source Xen [The13] from The Linux Foundation, Oracle VirtualBox [Ora13], and VMware Player [VMw13]. Amazon Elastic Compute Cloud (EC2) [Ama12] is perhaps the leading worldwide IaaS cloud.

The same classical businesses are copied onto the IaaS model. In other words, both physical servers and VMs can be remotely accessed via some remote connection protocol, namely SSH and Remote Desktop Protocol (RDP). Such approaches can be called Virtual Desktop Infrastructures (VDIs) and Virtual Private Servers (VPSes) under a virtualization context. An IaaS cloud provider is responsible for the security of the physical and virtual resources. Regarding the latter, isolation is a key component on multi-tenant clouds. VMs belonging to distinct customers can run co-resident over the same hypervisor and hardware, and thus attacks escaping the virtualization sandbox or exploiting a cross-VM setting endangers IaaS environments. Anyhow, a management interface functioning like a console is usually provided to perform all kinds of management tasks related with VM instances. Such interface may depict features of security,

[2] In the virtualization domain, snapshotting consists in copying VMs image files at some instant in time for backup reasons. Essentially, all persistent and volatile memory contents are dully stored. If desired, at a later time, that OS state can be easily recovered by simply reinstating those files.

namely related with cryptographic keys used to access each VM, making it a centralized point for the subscribed service. As a consequence, these interfaces comprise an interesting attacking point.

2.2.2 Cloud Deployment Models

It was previously said the NIST introduced four types of cloud deployment models. However, an additional one is also discussed throughout the literature, although being considered less frequently, named Virtual Private Cloud (VPC). From the customer point of view, all of the cloud deployment models can be summarized as follows:

- **Public Cloud**: A public cloud is perhaps the model that receives more attention. It is owned by a cloud provider that places it on some off-site data center or data centers. As such, management and security tasks fall to the cloud provider side which may delegate such responsibilities to a third-party organization. The services subscribed to public clouds are accessible through the Internet and the pay-per-use business model is used. Combined with the fact that public clouds are accessed by other customers, this model is a more risky one for its openness to the cyberspace and, thus, Service Level Agreements (SLAs) should cover security terms also.
- **Private Cloud**: In contrast with the public cloud model, private clouds are set up on-promises, within the trusted environment of an enterprise network and usually behind a firewall and other security controls. However, private clouds are more expensive since the owner has to buy the cloud infrastructure and hire a specialized technical crew to assemble the cloud and manage it over time. This model gives the enterprise total control over the cloud, which can be in charge of the corporation itself or a third-party one.
- **Hybrid Cloud**: The hybrid cloud deployment model embraces the previous two for exploiting the advantages and overcome the issues of each one. Some kind of specialized networking is required to link an on-site system with the off-premises cloud belonging to the cloud provider. Logically, the hybrid model extends the trusted edges of the perimeter, facilitating data transfers and communications from one point to another. The security risks of public clouds are hereby minimized. The costs of this model are somewhere between the costs of the public and private approaches, and it is thus placed on both on-site and off-site locations.
- **Community Cloud**: The community cloud deployment model concerns a particular cloud belonging to various parties, a committee of well defined entities that share some common goal. Each entity has access to specific resources of the cloud. The committee shares a budget for the cloud which may me managed by a third-party organization, if desired. Community clouds are placed both on-site and off-site. The security risks of this model

Table 2.1: Summary of the cloud deployment models with regard to ownership (Organization (O), Third-Party (TP), or Both (B)), management (O, TP, or B), location (Off-site, On-site, or B), cost (Low, Medium, or High), and security (Low, Medium, or High). This table appears in the second scientific publication [FSG+13b] of this dissertation.

Deployment Model	Ownership	Management	Location	Cost	Security
Public	TP	TP	Off-site	Low	Low
Private	O or TP	O or TP	On-site	High	High
Community	O or TP	O or TP	On-site or Off-site	High	High
Hybrid	B	B	B	Medium	Medium
VPC	B	B	B	Low	High

are also minimized, when compared to the public deployment model.

- **Virtual Private Cloud**: The VPC uses Virtual Private Network (VPN) connectivity and isolated resources on the cloud to create virtual private or semi-private clouds. Like a VPN builds upon other networks, the VPC seats on top of any cloud model previously described. Implicitly, VPCs are placed on both on-site and off-site locations, and are managed by the organization and the cloud provider, similarly to the hybrid model perspective. The most suitable example is Amazon VPC [Ama].

The cloud deployment models are summarized in Table 2.1. It is noticeable that each one has its advantages and disadvantages and, as such, cloud adopters should strive to assess what best fits their needs before committing to a specific one. This could be done by analyzing the purpose of the adoption of cloud solutions, the available budget for the short- and long-term, and security compliance with company policy. Nonetheless, Cisco [Cis13] admits that the hybrid cloud deployment model is the one that should be adopted from here onward, since it is the one that best adapts to enterprise computing and adheres to company policy. However, the advent of cloud computing made the industry spiral around various cloud computing approaches, and thus several miscellaneous cloud solutions came along, creating a diverse industry. As a consequence, more and more proprietary formats started to appear, resulting in vendor lock-in and delaying the construction of *interclouds*. Interclouds are thought as a network of interoperable clouds belonging to different providers that are able to speak in the same language, so as to facilitate data migrations and the development of intercloud connectivity. It is, therefore, required to streamline the cloud market into more standardized mainstream cloud computing.

2.2.3 Data Centers and Physical Security

In terms of IT rooms, cloud servers are a little different from their ancestors, namely mainframes and, more recently, distributed systems. In the past, a single mainframe computer would require an entire room to hold the components of the computer. Today, big IT rooms, spanning several square meters, can hold many physical and independent servers, that can be used in a standalone manner or coupled together to form a distributed system. Data centers is a common expression used nowadays to refer to the facilities housing such IT rooms. Some are

specifically built for clouds, thereby having in mind several aspects regarding the efficiency and security of the infrastructures. Firstly, the location of data centers is important. Since it can host many types of data belonging to high-value clients, it is crucial to build them on thoughtful locations, where earthquakes and other catastrophes are less probable, and that comply with political, governmental, and governance restrictions. Secondly, data centers may be designed to use the outside air to cool down the inner IT rooms and equipments, instead of installing air conditioners. This is called *free cooling*. By doing so, the owners of the data centers seek to achieve quality standards with respect to reliability, efficiency, uptime, and to Power Usage Effectiveness (PUE), which is a measure quantifying the energy efficiency. The PUE, combined with the *tier* levels, evaluate the quality of data centers. The closest to one the PUE is and the higher the tier level (the highest is 4 while the lowest is 1), the better, meaning that the data centers are highly efficient and sustainable.

Because of all of the above and also because clouds can have sensitive information, data centers are appealing facilities to attackers. They should thus employ both physical and logical security state-of-the-art techniques. Standard physical security controls, like surveillance cameras in closed circuit, are mandatory, particularly at access points, namely in passages that connect garages and main buildings, entrances and exits, and at strategic points to oversee particular IT rooms. Road barriers at the physical perimeter of the data center mediate car access, while roads built in curves instead of straight lines can difficult a potential physical assault on the facility. In extreme cases, barbed wire can be used on the perimeter. Regarding employees, there is always the threat of insider attacks. As such, unique identity cards and different factors of authentication should be put forth throughout the facility. For example, palm print scanners can be installed to open certain doors, only available to directors or administrators. In the case of the IT rooms and the clouds, only personnel with security clearances should have physical access to them. Cloud providers can also provide physical cages to protect the servers from prying hands, or even weighting chambers, useful for checking if any extra weight is detected, when exiting the rooms, due to stolen equipment.

For the logical security part, specialized monitoring crews oversee the network health status, looking into network links and bandwidth, servers, services, and particularly security events triggered by security controls spread throughout various network points. This is particularly important in large Internet Service Providers (ISPs), on which such duties are delegated to the teams of a Technology Operations Center (TOC), a Network Operations Center (NOC), and a Security Operations Center (SOC). The SOC monitors security events that are centralized into a Security Information and Event Management (SIEM) process constituted by several security controls like firewalls or Intrusion Prevention Systems (IPSes), and Intrusion Detection Systems (IDSes). The SIEM process involves aggregating all the event logs and segregating the ones of interest, correlating them using a large platform so as to understand potential intrusions or

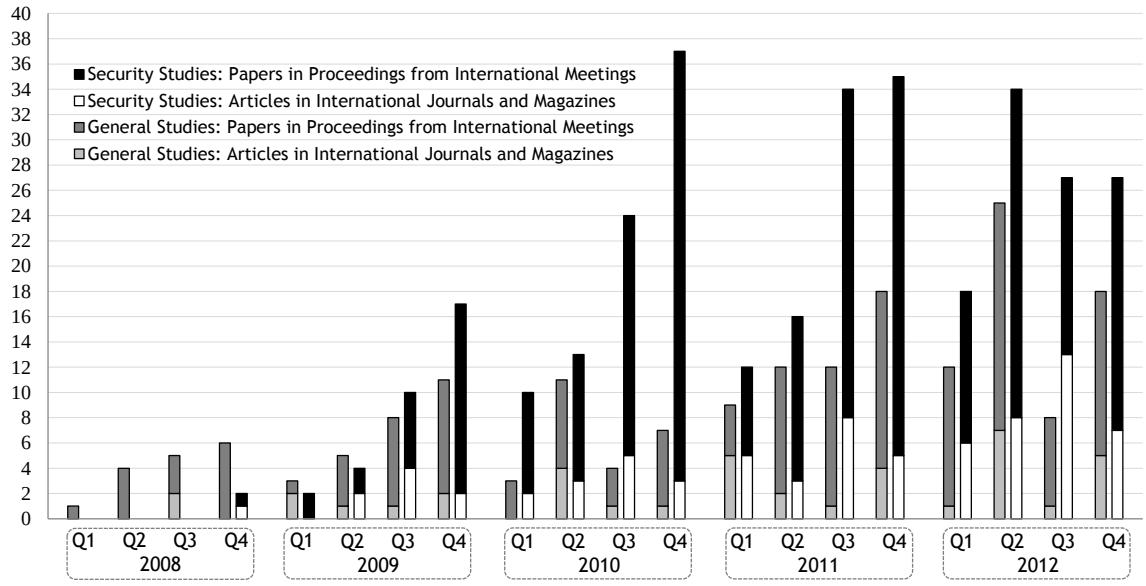

Figure 2.2: Chart depicting the analysis of the 2008-2012 time frame of the research trends in the cloud computing research field with respect to the number of papers and articles found in the main digital scientific databases and the quarter on which they were published. This chart was taken from the second scientific publication [FSG+13b] of this dissertation.

security issues.

2.2.4 A View of the Cloud Computing Research Field

Since the cloud computing field is relatively new, it is worth to find out how it evolved throughout the time while highlighting the security topic, since it is one of the main pillars of cloud computing. For this purpose, a study of the number of papers and articles published in the time period between 2008 and 2012 was carried out. The intent was to gather as many works on cloud-related topics as possible, while putting aside the ones related with security. The papers found were published on international conference proceedings, symposiums, workshops, congress or conventions, while the articles were published in international journals or magazines. These studies were divided into two main classes: *General Studies*, comprising works on any topic related with clouds; and *Security Studies*, comprising only studies on cloud computing security, service delivery models security, cloud deployment models security, vulnerabilities, threats, attacks, risks, security issues, and security solutions. Only the works with better and more appropriate contents were chosen, leaving aside those that seemed less interesting.

This study was included in the second scientific publication [FSG+13a] of the work presented in this dissertation. Its aim is to emphasize the importance of the research on the security side of cloud computing and show the trends exhibited over the years. The results of this study are depicted in Figure 2.2, on a graph starting in Q1 2008 and ending in Q4 2012. A total of 440 studies were gathered during this survey, of which 105 are the subject of journal and magazine articles, and 335 were discussed in papers included in conference proceedings. Studies not focused on the security topic comprised a total of 164 studies, while studies on

security sum up to a total of 276. As perceivable from the figure, in 2008, just a few studies not related with security were published on the general topic of cloud computing, since the research was just beginning. In 2009, the security topic gained momentum, and rose in terms of number of publications throughout the four quarters of that year. From then on, security on cloud computing has been one of topics receiving more attention, with a great number of publications. The highest peak on security publications is seen on Q4 2010. In particular, the number of journal and magazine publications on security has been steadily growing, reaching its highest value in Q3 2012.

Clearly, the academia is first concentrating its efforts in making cloud computing a safer place before starting to explore the great potentialities it provides. The study compiled only a small portion of the studies published on the main digital scientific databases such as ACM Digital Library, Elsevier, IEEE Xplore, and Springer, and thus it should be regarded as merely representative of the trends on this research field. Apart from the security topic, the other studies focused on, for example, mobile cloud computing, scientific cloud computing, eGovernment, green cloud computing, and performance optimization.

2.3 Classical Approaches to Authentication

Authentication is a two-party process for proving the identity of one entity to the other entity that checks the veracity of information exchanged by means of a protocol. The former is usually called the *prover*, while the latter is typically termed as the *verifier*. The verifier asserts if the prover provides the correct pieces of information, supposedly secret, for determining what action should follow. In case the prover indeed looks like being legitimate by correctly carrying out the authentication, the verifier typically allows access to some requested resource. Otherwise, if the verifier detects the prover cannot continue the authentication procedure, it probably denies access to resources and the authentication may be reset to the first step or subsequently denied.

In terms of computer systems, authentication can be performed in two main scenarios: it may be used to prove the identity of an entity to a local equipment, or it may be used to remotely assess if the transmitting party is who it claims to be in the context of the communication. In the most simplest model, local authentication is mostly done in an offline manner, by simply proving the combination of some username and password that, most likely, were set during the installation of a certain software. The software verifies the combination against all entries in a file stored locally and conveniently protected. If a match is found, the user is successfully authenticated.

Networking allowed to interconnect several computers and thus remote authentication was required. The principle is the same of local authentication, with the difference that the in-

formation is exchanged over the network. However, the evolution of the technology, data, and information has brought connectivity at an even lower cost. With this evolution, cloud computing shifted IT to remotely accessed clouds, migrating various kinds of applications and infrastructures, and therefore, authentication servers as well. This has brought the possibility of authentication over the Internet to services, but also the possibility of authentication locally but in an online manner. For example, Windows 8 supports login with an online Microsoft account or a standard offline account. Previous procedures that were used to locally authenticate a user may now be used to remotely perform the same task. This section elaborates on some concepts related with classical authentication methods based on passwords and describes some efforts in this field. In addition, cryptographic concepts related with this topic are also included.

Authentication by using passwords was used prior to computer systems. Actually, when two persons were to meet and each needed to verify the identity of the other, it was common to use two different passwords, one for each direction of the communication. When both entities simultaneously prove and verify, it is said that they are performing mutual authentication.

2.3.1 Passwords and Password Managing

Mostly in the seventies and eighties, computing was largely done by mainframes, on which text terminals possessing no local disk or memory would operate over such remotely located mainframes. Telephone circuitry would provide the means to dial-in to distant computers. At the time, passwords would suffice for authentication purposes, because the technology for carrying out brute-force attacks in feasible time was not yet available. Additionally, since the interaction with the machines was mostly text based, usernames and passwords comprised the most suitable means to achieve the aforementioned objective. In this sense, passwords could be simple, static, small-sized and easy-remembered (and thus low-entropy) words (*e.g.*, wyvern). Back then, the awareness on security was in its infancy. For example, protocols like `rlogin` and `telnet` exchanged data in plaintext, including the username and password combination for login. Eventually, SSH took over and it is regarded as a good protocol for remote operations. In the meantime, the security awareness has been progressively flipping to a positive side.

Today, passwords are used for almost everything in the cyberspace. The advent of an ever increasing Internet, associated with a diverse set of interoperable and connected devices has given rise to a large set of applications, web-based or not. As a consequence, system administrators and users are faced with the task of conveniently managing many distinct passwords for online and offline applications or devices. Some use the same password transversely, though it is considered one of the worst practices in this field and an approach more susceptible to malware. In the case of SaaS clouds, the problem gets magnified, since one requires an additional set of credentials for authentication per SaaS application. To address this management

problem, software was developed to manage passwords. This is the case of KeePass [Rei], an offline application that organizes password entries and safely encrypts everything in a single file. In turn, decrypting and accessing the database requires another (master) password. Moreover, password managers naturally were embedded to web browsers, due to their wide adoption, and additional features may be added to such thick clients easily. LastPass [Las] is another password manager featuring cross-platform support including mobile devices. For desktop computers, it is available as browser addons, and beyond automatically filling (browsers also support this) and submitting login forms, it may synchronize the passwords across all the LastPass devices.

2.3.2 Protecting Against Password Breaches

Although it was said in the previous subsection that security awareness has been flipping throughout the years, the massive increase of the Internet, and of web based applications, gave rise to a plethora of weak systems, vulnerable to some form of hacking, outpacing the rate at which the word about security spread. Weak frontend systems can allow an adversary to gain unauthorized access to potentially sensitive data, namely usernames and passwords. If the passwords are not conveniently salted[3] and hashed, the attacker thus can execute impersonation attacks.

For minimizing the danger of password breaches, the research community has focused on the particular method called *honeywords*. The term is inspired from *honeypot* systems, on which attackers are tricked or lured into thinking they have found some system of value, when in reality it is just a decoy. Honeywords were first introduced by Bojinov *et al*. [BBBB10]. They proposed Kamouflage, an architecture for building strong password managers. Kamouflage resists against offline password attacks by forcing an adversary to mount online attacks before learning any user password. This online approach eases the detection of potentially compromised accounts by detecting failed login attempts originating from a decoy mechanism that enlarges the password set with similar, but fake, passwords, statistically indistinguishable from the real counterparts. If one were to use random characters for the similar passwords, then it would be easy to detect such fake entries, from the attacker viewpoint. The concept was further enhanced by Juels and Rivest [JR13]. A honeyword system essentially functions like follows. The architecture requires always two components: one for storing the usernames and the passwords, and a secured separate one for storing the associations between usernames and passwords. If an intruder gains access to the list of usernames and passwords, he or she stills needs to obtain the correct matching between the two pieces of data, *i.e.*, the attacker needs to go online. In order for the statistically similarity to hold, a process $Gen(k, p_i)$, called chaffing, generates a randomly sorted list W_i of passwords with length k. An auxiliary and separate hardened component/system called *honeychecker* maintains an association $c(i)$ of the random index of the legitimate password p_i of the user u_i in W_i. The authors defined various

[3]Salting passwords is the process of concatenating random bytes to a password for subsequent hashing. Such scheme maximizes security against brute-force attacks by means of rainbow tables.

methods for chaffing, and discussed their advantages and disadvantages, their security, and the implications of implementing them in user interfaces. Note that hashing and salting parameters are abstracted in the proposed protocol.

2.3.3 Web-based Sessions

The web and HyperText Transport Protocol (HTTP) grew alongside with the Internet. Documents statically stored on web servers would provide information publicly to someone far away, connected via the Internet to the server. Since then, the client-server paradigm has evolved, although other distributed models for communications have been created, such as Peer-to-Peer (P2P) networks. The explosion of the Web 2.0 triggered developments in dynamic web pages. Eventually, some became online applications accessible over the Internet, requiring users to register to the application and use the chosen credentials for subsequent authentications. SaaS cloud computing has strikingly adhered to this for authentication.

But, web-based communications are asynchronous and the underlying HTTP is stateless. For applications requiring some sort of session in an authenticated way, it was required to exchange a token for every request. This is the case of web *cookies*, designed to be a reliable mechanism for websites to remember authenticated users with stateful data. In part, third-party tracking cookies have brought privacy concerns since they can register user activity on the browser. The problem is that cookies can be copied from one browser to another and continue to be used as normally. Web servers do not record information about the origin of the requests, and thus using cookies from different IP addresses at a time is legit. It is up to the application logic to implement security measures, such as risk-based authentication (this is further discussed in the next section). However, where no conditions or clauses for authentication are set, cookies are vulnerable to cookie theft through MitM attacks or MitB by means of malware (*e.g.*, malicious addons), even for HyperText Transport Protocol Secure (HTTPS) connections. For unencrypted wireless connections, a simple traffic sniffer would suffice for capturing communication packets passing by. In this scenario, malicious MitM applications could just sniff the HTTP cookies and carry on session riding. For example, Facebook did not used HTTPS in its early days, and it was popular to steal cookies and browse profiles belonging to other people nearby, connected to the same wireless network.

2.3.4 Protecting Against Session Riding

To reduce the risks posed by HTTP cookies, Google proposed an extension for the Transport Layer Security (TLS) protocol as an Internet Engineering Task Force (IETF) draft [BH12], called Channel ID. It is described as follows. A server cryptographically bounds authentication tokens to underlying TLS communication channels, rendering token theft fruitless via MitM attacks, assuming a nondisclosure of the server private key. A new handshake message type is proposed in the draft for the purpose, after the server sends its certificate. An elliptic curve Data Signature

Algorithm (DSA) signature belonging to the client corresponding to the private key of the TLS Channel ID is sent to the server, therein tying tokens to specific client TLS sessions—the Channel IDs [Bal12a]. This assures tokens are only ever used over TLS channels authenticated with specific public keys belonging to the clients. The mechanism is, nonetheless, still vulnerable to MitB attacks because malware can use the same token-bounded channel. This protocol was under testing on the Google Chrome browser, on which some issues were found [Bal12b] by the authors. They subsequently amended them, and thus this protocol seems adequate and will probably be used in the future.

2.3.5 Cryptography in Authentication

Cryptography is utmost importance for the security of communications in both local and remote connections. The powerful mathematical theories behind modern cryptography provide a wide spectrum of applications. Classical cryptographic mechanisms were mostly used to fulfill confidentiality requirements but, nowadays, cryptography provides the answer to anonymity, encryption key exchange, digitally signing, integrity, digital coins, non-repudiation, among others. While symmetric-key encryption algorithms are used to safely store files and protect communications over potentially insecure channels, the Diffie-Hellman protocol over any cyclic group, in which the Discrete Logarithm Problem (DLP) holds, can be used to set the encryption keys for the session efficiently and without the requirement of having a pre-established secret between the entities, over an insecure channel and under passive MitM. On the other hand, trapdoor functions like the Rivest, Shamir, Adleman (RSA) provide for the means to sign messages with a private key, which can be verified with the associated public key by any entity. The signing mechanism can be immediately applied to authentication, since the prover may demonstrate that he or she owns a given private key by simply signing a random challenge from the verifier. If the verifier is sure that a given public key belongs to the prover, than he or she only needs to verify the signature to know if it was the rightful holder of the private key that signed it, assuming that the key pair was not compromised.

A problem that may arise when using public-key cryptography is that of trusting a certain public key. Digital certificates come to overcome this problem. A digital certificate tags along with the public key a digital signature of another entity, which is superior in hierarchy, demonstrating the trust in that public key. In other words, a third-party entity certifies that a certain public key is indeed associated with that certain entity. But then, the same problem of trust arises for the public key belonging to the third-party entity that signed the certificate. In this case, another certificate comes along validating that public key, and so on. This is called the *certificate validation chain* and involves the definition, implementation and usage of a Public-Key Infrastructures (PKIs) for managing certificates. The certificates in the top of the chain are highly trusted, belonging to well regarded companies (*e.g.*, VeriSign), or Certificate Authoritys (CAs), and are self-signed. Digital signatures depict the non-repudiation property, which means that

the entity associated with a given public key cannot deny that it signed a certain message. Public-key cryptography and particularly digital signatures can be directly applied to authenticate an entity or even a message, since the private key should be in sole possession of the rightful owner.

On the other hand, Zero-Knowledge Protocol (ZKP) (*e.g.*, Schnorr Protocol) aim at probabilistically proving the possession of some secret by the prover, while exposing as few information as possible on the communication channel or to the verifier. ZKPs recur to challenge-response interactions and also to mechanisms of public-key cryptography. Normally, the bigger the challenges, the slimmer the chances for adversaries to guess the correct secret. Moreover, ID-based cryptography [Sha85] (invented by Shamir, one of the authors of RSA) is an approach that utilizes public information of entities to derive a public key. Such public information include the email address or domain name encoded into an ASCII string. Message Authentication Codes (MACs) are used for authenticating the origin and the integrity of a message, rather than an entity. So, authentication has more to it that at a first glance. In communications, the initial step should always be the validation of the other end of the line, but in the long-term it is equally important to continuously verify the communications as in web-based sessions.

2.4 Novel Approaches to Authentication

In the last few years, the industry and the academia started moving into new authentication approaches, because password-based authentication could no longer stand against the myriad of technology and algorithms available for cracking and testing passwords. This section looks into some of those rising trends. Although the concept behind each method may date several years back, only now the awareness and the need for stronger authentication, combined with new technology has given enough motivation to shift from outdated schemes to more robust mechanisms. Chronologically, the Internet exploded first in the beginning of the nineties, then came the Web 2.0 near the end of the millennium and, in the last handful of years, the mobile technology exploded along with cloud computing. It is a natural step to adapt to such trends and change authentication procedures accordingly. Particularly, mobile technology has eased such a process with the rise of the Bring Your Own Device (BYOD) paradigm, though it has brought some security issues as well.

Perhaps the main organizational bodies behind the force driving authentication trends are the Fast IDentity Online (FIDO) alliance [FID] and the Initiative for Open AuTHentication (OATH) [OAT]. The former is constituted by players like Google, BlackBerry, PayPal, and Yubico, while the later comprises vendors like Symantec, VeriSign, and SanDisk. There is a clear diversity of industry vendors composing these organizations, namely SaaS service providers, smartphone sellers, hardware manufacturers and security specialists. The reason for such a diverse portfolio is that authentication is inherently becoming less abstracted on the underlying

hardware, and is increasingly converging to device-centric and user-centric models. The technologies and mechanisms of several fields converge to build novel authentication approaches. Starting off right from the hardware and moving up the stack to the user and the application layer, in a distributed manner by resorting to additional layers of security and commodity, delivers a seamless user experience with regard to authentication.

2.4.1 Single Sign-On

With the rise of cloud computing came the massive use of SaaS applications. As previously said, managing many credentials for each of the SaaS application may comprise an arduous task. For overcoming this problem, the industry has came up or is in the process of largely adopting the Single Sign-On (SSO) approach. SSO is a process that permits a user to access multiple applications by simply authenticating once on a centralized and online service, whose owning entity takes the role of an Identity Provider (IdP), the *asserting parties*. IdPs vow to the users for holding their credentials, who assume it acts honestly. Essentially, a user authenticates to one system and, from there onward, it is not needed to authenticate again in the applications that are bound to that particular IdP, called *relying parties*. For this to work, each application is requested to make a few code modifications to the register and login forms so as to support authentication by the IdP. The user must also possess an account on the IdP. When a user wants to authenticate to a certain application, it chooses the IdP desired. The application communicates with the IdP for assessing the identity (here, identity refers to the information uniquely identifying a user, such as the email or username) of the user. If the user has not already established a valid HTTPS session with the IdP, meaning that authentication has already been made and a cookie has been issued, it then authenticates normally to the IdP. The IdP then issues a security assertion to the target application, certifying the user is indeed associated with a certain identity. The target application takes that as proof and accepts the identity given by the IdP. For example, Google implements SSO in its products, allowing users to swap between applications (*e.g.*, Gmail and YouTube) seamlessly while maintaining authenticated sessions. Several well-regarded online services (*e.g.*, StackOverflow) allow to login by authenticating with Google instead of requiring to register a new account.

In order for the target applications and IdPs to speak the same language, a protocol is required. The *de facto* standard for the SSO process is the Security Assertion Markup Language (SAML) [SAM05], an eXtensible Markup Language (XML)-based open standard for exchanging authentication and authorization data. SAML belongs to the Organization for the Advancement of Structured Information Standards (OASIS) and the most recent version dates to 2005 (version 2.0). In the particular case of SAML, the phase of authentication to the IdP falls out of the scope of the protocol. It is up to the IdP to implement any desired mechanism, including linking the IdP to some local directory service. This is important for enterprises, since their internal applications are usually authenticated by utilizing directory protocols like Lightweight

Directory Access Protocol (LDAP) or Active Directory (AD). In such case, the IdP may very well be the company. The McAfee Cloud Single-Sign On solution [McA13a] allows to place an authentication point at network borders and mediate access to external SaaS applications by means of internal domain usernames. Other SSO protocols include the OpenID, which is on version 2.0 and whose specification is at [Ope07] and, more recently, the Mozilla Persona, whose underlying protocol is BrowserID [Moz]. Both follow the same basic operation of SAML. What may change is how they do it and the kind of security employed. For example, the OpenID protocol specifies an agreement of a symmetric key by means of the Diffie-Hellman key agreement protocol between IdPs and relying parties, so as to exchange authentication data securely. On the other hand, the BrowserID protocol utilizes public-key cryptography and certificates.

2.4.2 Multi-Factor Authentication

Multi-Factor Authentication (MFA) enhances authentication by adding additional layers of security to one-factor systems in order to further verify the identity of some prover. However, those layers, or factors, should all be distinct and of different type, otherwise little or no security would be complementarily achieved. The factors are usually categorized into one of the following classes [Kir13]: *something you know*, *something you have*, *something you are*, and *something you do*. These classes define a given MFA process, and can be combined as desired. The *something you know*, for instance, adheres to password-based authentication, because the prover authenticates by providing *knowing* some secret information. The *something you have* requires the possession of some software or hardware token, like a certificate or a smartphone, respectively. The *something you are* calls for biometric technology for reading some unique biological signature, namely the fingerprint, the DNA, or the retina. Finally, the *something you do* includes speaking predefined voice patterns or responding to challenge response schemes.

2.4.2.1 Multi-Factor Authentication Versus Multi-Factor Authorization

Sometimes, authentication is confused or mixed up with *authorization*. Authentication and authorization are, in fact, orthogonal to each other in a complementary sense. Authorization is the process of permitting or granting some entity access to some particular set of resources on the behalf of the original owner. It complements authentication in a way that proving and verifying an identity is the first stage for authorization, which may come as a second step on an arbitrary number of times. The most simplest form of authorization may be illustrated with the Windows User Account Control (UAC). If UAC is turned on, it will ask permission to the user for software to make changes to the computer. But, for good reason, authentication and authorization may be merged, since the gap between the two is not crisp enough in some online systems.

Social networking, data mashups[4] and connectivity have provided the means to integrate distinct third-party resources. But such an integration requires some sort of authorization. Initially, big players developed their own proprietary protocols to address authorization requisites. For example, Google deployed AuthSub [Gooa], while Yahoo developed Browser-Based Authentication (BBAuth) [Yah]. The OAuth [OAu] (not to be confused with the OATH organization) specification comes as an open protocol to merge the wisdom of such well established proprietary authorization protocols so as to make a secure and simple standard method for web, mobile and desktop applications. It allows a third-party to obtain limited access to an HTTP service. Examples of this scheme includes third-party applications fetching Facebook profile data or other services peeking into Google contacts. In the realm of authorization, OAuth issues tokens or tickers in exchange for user credentials that allows other entities called *consumers* to access private resources on the behalf of the user. It can use HTTP basic or digest authentication nonetheless. Google, Yahoo and other services discontinued their protocols in favor of open standards, recommending the use of the OpenID and OAuth hybrid combination.

Kirkland, at his talk at the Linux Foundation Collaboration Summit of 2013 [Kir13], argued that, similarly to what was discussed above, authentication is somewhat in the gray area together with authorization. Thus, MFA also falls into the pit. By having this in mind, it is entirely possible to come up with the term *multi-factor authorization*. For models adopting hybrid combinations of protocols, like OpenID and OAuth, both MFA and multi-factor authorization can simply be called *muti-factor auth* until their differences are not endorsed further or, in fact, until their contrasts vanish and they merge into unified protocols. The discussion on the remaining part of this section is mostly focused on MFA, but it does not dismisses multi-factor authorization for added value in the discussions.

2.4.2.2 Two-Factor Authentication

One particular instance of MFA is known as Two-Factor Authentication (2FA), a popular authentication framework used nowadays [GU13]. 2FA has been widely adopted by major Internet players, namely Google, Apple, Facebook, PayPal, Amazon, and others. Industry appliances for enterprise environments are also available in the market. This is the case of RSA[5] Authentication Manager [RSAa], an appliance for managing 2FA of the RSA SecurID token (this solution is detailed in the subsection 2.5.4.1). Such appliances are placed at the network edges, as firewalls are, for the 2FA to be integrated within remote VPN authentication, for instance. 2FA has been around for decades, more precisely since the eighties, but only now the technology and motivation behind the scenes supported mass-market deployment.

[4] A data mashup is the integration of data and applications from multiple sources, namely of web and SaaS origin. For enterprises, this is useful for business analytics and, more recently, big data.

[5] In this case, RSA is not related with the RSA algorithm, but with the persons who invented it and founded the RSA Security subsidiary of the EMC corporation.

2FA is typically designed to take advantage of the combination of *something you know* with *something you have*. A common application of this in daily life is in Automated Teller Machines (ATMs); whereas the banking card is the *something you have*, the associated Personal Identification Number (PIN) is the *something you know*. However, PINs are pieces of static information, like passwords. For online authentication, the first factor authentication is typically the combination of username and password, while the second factor is the so-called One-Time Password (OTP). OTPs are generated on additional separate systems, and can be transmitted by an out-of-band channel like the Short Message Service (SMS), or can be generated in an offline manner on physical or software tokens. The out-of-band method is adopted by banks for confirming online transactions. Note that this approach is a mixture of authentication and authorization, because not only users prove having the smartphone around for receiving SMS messages and insert the correct OTPs, but also for authorizing a specific operation in real-time. Although 2FA does not completely solve authentication issues, it plays an important role in the fight against online fraud.

2.4.2.3 Approaches to Two-Factor Authentication

In 2006, GPayments, a company focused on authentication and payment solutions for online transactions, summarized several 2FA technologies from an enterprise perspective [GPa06]. That set of technologies included hardware and software tokens, smartcard readers, chip-enabled Universal Serial Buss (USBs), Transaction Authentication Number (TAN) lists, mobile phone SMS or application, and terminal profiling.

Smartcards (*e.g.*, Cartão do Cidadão in Portugal) are capable of securely storing cryptographic objects in non-volatile memory and perform calculations on a microprocessor embedded in the card. Typically, they receive a random challenge for signing purposes within the confinements of the card, proving the authenticity and integrity of the information, and further authenticating an identity. This concept can be incorporated into USB dongles (*e.g.*, RSA SecurID 800 Hybrid Authenticator, an USB stick with offline OTP generation and cryptographic capabilities), therein making this method widely compatible with any USB-ready device. TAN lists are simply composed by OTPs printed on paper that are used to access protected resources. When an OTP is used, it becomes invalid, and so on until the list runs out. Terminal profiling consists in gathering utilization characteristics and use them to limit the usage behavior of certain accounts. Subsequent sessions would be limited, but with a certain extensibility and flexibility, to such a profile. Although its effectiveness may be low yielded and a profile is producible, it is inexpensive and can be deployed as a backup mechanism. In fact, various services now come packed with such a system, such as Google or Facebook, which learn user habits and warn them if an outlier is detected from the normal pattern. The term for calling this approach is *risk-based authentication*, which is briefly overviewed in subsection 2.5.3.

Several years back there was a lack of transparency and openness in the authentication market, hindering the adoption of MFA. On the one hand, smartcard- and USB-based technology is more secure, but on the other, it is more costly to deploy and also to maintain. Additionally, the durability of hardware tokens may not be that long. As such, it may be required to replace them every couple of years or so. Today, a few Request for Comments (RFC) form the basis for the adoption and implementation of 2FA, being the smartphone the preferred device for receiving SMS messages or for generating OTPs by means of a software token.

2.4.2.4 The Somebody You Know Factor

Brainard *et al*. [BJR+06] proposed another factor of authentication through the concept of *somebody you know*. The *somebody* term replaces the *something* in the conventional 2FA approach, and refers to the social network of a user. It is based on the notion of vouching, which is a peer-level human-intermediate, and its purpose is to use it on emergency authentication procedures, whenever a primary authenticator is unavailable. The authors presented a prototype based on RSA SecurID.

The *somebody you know* approach is explained as follows. An helper relation $H(X, Y)$ between an helper X and an asker Y is *a priori* publicly enrolled in the system. An authentication ceremony, denoted by AC, is a sequence of interactions between various parties with certain relations among them, may such parties be software agents or persons. $AC_T(P_1(p'_1), P_2(p'_2), ...)$ consists in an authentication ceremony of some type T of the parties P_i with identifiers p_i. When running the protocol, the helper X authenticates the asker Y by alternative means like voice or face recognition. X normally authenticates to server S, which asserts X is indeed enrolled to help Y and creates a temporary vouching session. S then passes an asker-specific, ephemeral alphanumeric vouch-code VC with a small amount of entropy to X, in turn verbally transmitting it to Y. This assumes that, for a given ceremony $AC(P_i(p_i))$, all parties act honestly. Then, Y enters the vouch code along with the username and password pair to authenticate. If unsuccessful, the vouching session is aborted, otherwise a temporary password is created for subsequent sessions and the vouch code becomes invalid.

In the protocol, an adversary has a very small probability of presenting both the first factor information as well as the vouch code. Furthermore, the entire protocol is based on trust, but as in the peer-level Web of Trust defined fo Pretty Good Privacy (PGP).

2.4.3 Industry Protocols and Algorithms

Below are detailed descriptions of the protocols used for the generation of OTPs, proposed by the OATH. Other protocols and algorithms devised by the FIDO alliance and the OATH organization are also summarily explained.

2.4.3.1 One-Time Passwords

Two IETF RFC are particularly relevant in the 2FA discussion, because 2FA largely depends on embedding efficient algorithms for generating OTPs on mobile devices or small portable tokens. The two RFC define two types of OTPs: a counter-based HMAC-based One-Time Password (HOTP) described in RFC 4226 [MBH+05] and a Time-based One-Time Password (TOTP), described in RFC 6238 [MMPR11]. The authorship of both algorithms if of the OATH. The TOTP algorithm is an extension of the HOTP algorithm.

In essence, given a shared secret K and an eight-byte counter value C, an HOTP value is obtained as shown in the expression (2.1):

$$\text{HOTP}(K, C) = \text{Truncate}(\text{HMAC-SHA-1}(K, C)). \quad (2.1)$$

The 20-byte hash value of the Hash-based Message Authentication Code (HMAC) with the Secure Hash Algorithm-1 (SHA-1) as the hash function (as defined in RFC 2104 [KBC97]) is truncated and transformed into a reasonable sized numeric value according to a system parameter $digit$. The $digit$ parameter defines the final size of the HOTP value, and commonly is a six or eight digit number, so as to favour the user experience, taking out the burden of having the user typing too many digits. Basically, the truncate operation randomly selects an offset by filtering and transforming the low order four bits from HS into a number, where HS is the 20 byte hash value of HMAC-SHA-1(K, C). By design, the offset cannot exceed the number 15 and, thus, a four-byte dynamic binary code starting at the offset position is extracted from the HS string without overflowing. Finally, the binary code is transformed into a number $Snum$ and is subjected to a modulo operation to generate the final HOTP value. Equation (2.2) mathematically expresses the reduction modulo 10^{digit}:

$$\text{HOTP} = Snum \mod 10^{digit}. \quad (2.2)$$

The TOTP algorithm may replace SHA-1 with the SHA variants of 256 and 512 bit hash output, SHA-256 and SHA-512, respectively, or with other cryptographically secure hash algorithms in the future. The TOTP replaces C in equation (2.1) by T, an integer representing the number of time steps between an initial counter time T_0 and the current Unix time. Formally, for a specific time-step X, meaning that TOTPs change every X seconds, T is computed as shown in expression (2.3):

$$T = \frac{(\text{Current Unix Time} - T_0)}{X}. \quad (2.3)$$

Typically, a short value of 30 or 60 seconds is assigned to X as a balanced measure between

usability and security. Moreover, by default, T_0 is zero, and the dividend represents the Unix Epoch time. In both algorithms, C and T are called the *moving factors*, providing them with the freshness for each iteration.

2.4.3.2 The Online Security Transaction Protocol

One of the key goals of the FIDO alliance is to make the Online Security Transaction Protocol (OSTP) an internationally, openly recognized standard body, ubiquitously available across devices for strong authentication [FID]. Achieving such goal and making such a protocol stack requires commitment from the industry, ranging from hardware chipset vendors to backend server manufacturers, turning such a coordination into a complex affair.

The protocol has an umpteen number of applications, being the initial focus on accessing Internet services through web browsers and web applications. Other possible deployment platforms include authentication in OSes at the login or lock screens. It is designed to transversely support a wide spectrum of technologies of the various factors in MFA. In other words, it can be deployed in traditional OTP mechanisms, biometric technology such as fingerprint scanners, voice and facial recognition tools, standard PINs, USB tokens, or even Trusted Platform Module (TPM)[6] implementations.

A software is required to be installed on end user devices so as to process information about their biometric hardware or TPM capabilities. A cryptographic process is applied to generate a secret, shared between devices and backend authentication servers. In contrast with TLS/ Secure Sockets Layer (SSL), the protocol does not assume a pre-trust relationship. It rather enrolls devices and creates the trust by provisioning the shared secret. The protocol is resistant against MitM, phishing, and replay attacks, but requires to be more refined and matured for a broad adoption in the industry. It represents, nonetheless, a great step towards next generation passwordless authentication.

2.4.3.3 The OATH Challenge-Response Algorithm

The OATH Challenge-Response Algorithm (OCRA) is defined in RFC 6287 [MRB+11] and is part of the OATH algorithm portfolio to facilitate the democratization of strong authentication throughout the Internet and back off of proprietary schemes. A challenge-response scheme is a method suitable to maintain an asynchronous authentication system, but an open scheme may further provide the building block for interoperable and interchangeable authentication systems. Such

[6] TPM is a specification for building secure cryptoprocessors that can store cryptographic material on the hardware chip for added protection against malware, for instance. TPMs are embedded into motherboards on desktop and laptop computers.

an asynchronous system may prove useful for various use cases where encryption is not possible, and provides resiliency against replay attacks, because intercepting a challenge-response pair is useless since the challenge is a one-time randomly generated value.

The OCRA generalizes the HOTP algorithm, and offers one-way and mutual authentication, along with digital signature capabilities, with extended functionality. As in the HOTP algorithm, let K be a shared key, expressions (2.4) and (2.5) define how an OCRA response is calculated by a prover along with the DataInput variable, respectively:

$$\text{OCRA} = \text{CryptoFunction}(K, \text{DataInput}), \tag{2.4}$$

$$\text{DataInput} = \{OCRASuite \mid 00 \mid C \mid Q \mid P \mid S \mid T\}. \tag{2.5}$$

DataInput is a set of security parameters concatenated as in expression (2.5), in that order, including the 128-byte challenge Q previously sent by a verifier. While 00 is simply an empty delimiter byte, C is a counter, P and S are variable in length, each representing an hash value of a PIN or password known to all parties and a UCS Transformation Format–8 (UTF-8) encoded string with session information (*e.g.*, TLS session identifier), respectively, and T is the Unix Epoch timestamp. $OCRASuite$ is a parameterized string that defines the version of the algorithm (version 1 only so far), a template HOTP-H-t, where H is an hash function that extracts a dynamically truncated t-sized number from HMAC output, and which options and their arguments are to be used for the aforementioned parameters. The HOTP-H-t template determines which algorithm family is used and what the CryptoFunction does. Supported hash functions are SHA-1, SHA-256, and SHA-512. This setup resembles the SSH session negotiation phase protocol.

The above defined computation can be used with three OCRA modes for authentication: one-way challenge-response, mutual challenge-response, and signature. The one-way mode is typically used for authenticating a client prover, while a server assumes the verifier role. While in the one-way mode the verifier first generates the challenge, the protocol is first initiated by the prover in the mutual authentication mode in order for the client to be assured that it is talking with a valid server. In the mutual case, two challenges QC and QS and two responses RS and RC are computed, while the challenges in this mode can be computed differently by the prover and the verifier for each of their responses. Technically, this means that each party can define its own $OCRASuite$ string, just like SSH. Both the one-way and mutual authentication concepts can be adapted to produce a third mode, the signature mode. Basically, the difference from this mode is in the challenges that are sent by the verifier, which are either the data to be signed or derived from the data to be signed using an hash function, for instance. These are called signature-challenges, and the prover then replies normally with an OCRA value if the one-way mode is used. For the mutual mode, the prover first sends a typical challenge, to which the verifier responds along with the signature-challenge. The prover finally

responds. In mutual authentication, a pair of pre-shared keys can be used to further enhance the strength of the algorithm.

The security of the OCRA relies on the assumptions provided by the underlying HOTP algorithm. In addition, implementation of the algorithm should take into consideration possible replay attacks when optional parameters are skipped in the signature mode. For example, a signature value can be computed as defined in expression (2.6) when the remaining parameters are set to optional:

$$\text{OCRA} = \text{CryptoFunction}(K, QS). \quad (2.6)$$

Then, if the same data or hash of the data equals QS some other time in the future, the response signature matches the same one. Moreover, RFC 6287 does not say when HOTP-H-t is agreed in the protocol. It is, therefore, assumed that these parameters are agreed sometime before the authentication phase, just like what happens with the shared key K.

2.5 Authentication State-of-the-Art in the Academia and in the Industry

New computing environments and technology open up new means for devising novel and alternative authentication mechanisms, or making classical ones more robust. The case of cloud computing urged the need to harmonize authentication by eliminating the complexity and burden of managing authentication for an umpteen number of SaaS and PaaS applications, IaaS management interfaces, and remote VMs. The emergence of the mobile era is also revolutionizing the field of computer based authentication. Within these lines of thought, both the academia and the industry have focused on enhancing the security and the user experience for carrying out authentication. Below are overviewed research advances from the academia and some noteworthy solutions deployed by the industry.

2.5.1 Quick Response Codes

Quick Response (QR) codes became useful to authentication because mobile devices came along, conveniently incorporating powerful cameras capable of easily reading them. QR codes are two-dimensional barcodes invented in the nineties, which encode pieces of information on a square grid using black and white boxes. They are now being widely deployed throughout the world, but was initially very popular in eastern markets, namely Japan. They offer high-speed readings and contain duplicated blocks for redundancy purposes, helping in error-correction. They are limited in the amount of information that can be encoded and are mostly used for encoding an Uniform Resource Locator (URL) or pieces of cryptographic information.

In the literature, QR codes are commonly used for MFA schemes. Liao and Lee [LL10] devised

a 2FA passwordless scheme by encoding nonces in QR codes. The registration phase exchanges the user identity and a hash h of both the concatenation of the secret of the verifier and that identity. In the authentication phase, the prover demonstrates it is capable of decoding a QR-encoded challenge that contains the result of a XOR of a random number r with h. The random number r is possible to obtain by XORing the entire challenge back with h again. If the verifier asserts that the received r is the correct one, the user becomes authenticated, because it proves having h, previously calculated. This is a simple mechanism requiring some sort of medium to store h safely, namely on a smartphone. Similar works [MA11, CLJ+11] extended this method to anti-phishing SSO models. Others [LKL+10] explored the possibility of utilizing QR codes for certifying online banking transactions, while relying on a trusted third-party entity for OTP generation. In 2012, Google tried [Nar12] to strengthen authentication to Google accounts. The principle was to scan QR codes of web pages and then automatically login through the browser, assuming the smartphone decoding the barcode securely communicated with the servers to supply identity information and prove the legitimacy of the login. However, this project was put away. Moreover, Perez *et al*. [PCG+13] proposed unlocking computers in an offline manner with QR codes and public-key cryptography. The computer generates a key-pair and QR-encodes them for the user device to read and store while saving the public key for post authentications. When locked, the computer shows a QR code with a random challenge. For unlocking, the device reads it, decodes it, signs it, and finally QR-encodes it for the computer to verify the signature and proceed accordingly. The computer must incorporate a camera for scanning the signatures encoded into QR codes.

2.5.2 Multi-Factor Authentication and Cryptography

MFA is an abstract procedure mixing four main conceptual pillars independent and different of each other. MFA mechanisms joining two factors of the same origin gives no additional security beyond what the static one-factor login systems give. With this in mind, research on MFA schemes is mostly oriented for 2FA. Some works simply focus on the second factor, leaving the first factor to password-based authentication, while others avoid passwords entirely.

Choudhury *et al*. [CKS+11] described a strong scheme by using smartcards and an out-of-band second factor specific for cloud computing. The smartcards require to be especially designed for the protocol, in order to store specific information, and a registration phase is necessary before issuing of the smartcard for the user, since the data has to be installed in them. The execution of the protocol provides session key establishment and mutual authentication. It is resistant against MitM, replay attacks and Denial of Service (DoS) attacks. Pu [Pu10] showed how to improve a 2FA scheme based on smartcards against key-compromise impersonation when the verifier long-term key is compromised. Their scheme also computes session keys.

Other works have focused on the biometric factors. Yassin *et al*. [YJIZ12] utilized the Schnorr

digital signature scheme with fingerprints for building an anonymous password scheme. In the registration phase, the authors isolated unique features from the fingerprint to send to the verifier, which sends back an encrypted file containing credentials that is required to be stored on a USB stick or in a smartphone. Afterward, the file is decrypted and is exchanged as a first factor, followed by the second factor comprised by the fingerprint. Moreover, Yassin *et al.* [YJI+12] used password-based authentication mixed with ZKP for achieving anonymous logins in regard to privacy-preservability in cloud computing and for mutual authentication. Although ZKPs are computationally heavy, their scheme showed to be sufficiently efficient. Grzonkowski and Couglin [GCC11] extended a ZKP of their own for cloud-based services. It uses passwords and overcomes some flaws from Kerberos so as to provide strengthened security on an SSO scenario over the Internet. Furthermore, cryptography is also on the basis for other approaches, namely ID-based authentication, which resorts to public identity data for deriving cryptographic keys. Yang and Chang [YC09] proposed an ID-based mutual authentication scheme with key agreement under Elliptic Curve Cryptography (ECC). Other studies [YY09, ClYS11] have enhanced it to cover security pitfalls of insider attack, impersonation and perfect forward secrecy. Jaidhar [Jai13] has also provided an improved version of a time-bound ticket-based mutual authentication scheme for cloud computing, first proposed in [HZ11], which uses clients smartcards for issuing digital tickets.

2.5.3 Risk-Based Authentication

In simple terms, risk-based authentication consists in making an access control decision based on the typical behavior found for a certain user. This approach emerged from the necessity of distinguishing between legitimate and illegitimate authentications. For example, for users not leaving the home country often, or at all, the source IP address will most likely be a dynamic or a static one assigned by the ISP if the user is at the home network or at the enterprise network, respectively. If an authentication is detected from overseas countries, that may mean that the account got compromised and someone is trying to login from an unusual location. Risk-based authentication also involves other components, namely the client agent used for authentication, like a browser or a smartphone. Mechanisms for making the access decisions may resort to artificial intelligence techniques, namely Bayesian networks [Lai08] and fuzzy logic [HAKH10]. Some [TWO+12] even monitor mouse and keystroke dynamics, and others [RQSL12] use external sensors to monitor the physical behavior of the user habits, while deciding if authentication is required or not. The RSA division of EMC provides [Cur13] an industry manager with risk-based authentication capabilities.

2.5.4 Industry Solutions for Authentication

MFA solutions have been quite diversified throughout the industry, evolving at a steady pace. Today, there are a few solutions worthy of mentioning. Each one varies in security, cost, and usability to some degree, but most of them elaborate on two factors only. This subsection

describes below some of those approaches.

2.5.4.1 RSA SecurID

One solution that is worth to describe in the context of 2FA is the RSA SecurID [RSAc]. RSA SecurID can be viewed as the 2FA pioneer in terms of hardware-based token generators. It contributed to the development in the area, opening way for other emerging solutions in the market. The logic behind the 2FA method in SecurID is, nevertheless, slightly different from other popular 2FA solutions. Instead of entering the second factor code after the username and password, it is attached to the end of the password in the standard login form.

The algorithm is described as follows. The token is configured with a factory encoded 128-bit seed—a random key unique to each device. The key, together with the current time, are inputted as a large number to an hash function that digests them down to a six digit output representing the TOTP. Every 30 or 60 seconds, the code changes. Conceptually, the code is synchronized across each device and with the backend servers, which have to store the aforementioned keys in order to verify each OTP. Although not often, the clocks on the devices might sometimes drift backward or forward, creating a time-skew between devices and servers. That drift is called token offset, and can be detected by the servers at login requests. If a TOTP mismatch is detected at a given time t, the server then generates the previous and the next TOTPs for the times $t - 60\ s$ and $t + 60\ s$, respectively, for an algorithm adjusted with $X = 60$ seconds. If the TOTP provided by the user matches one of the $t - 60\ s$ or $t + 60\ s$ TOTPs, the server adjusts the clock for that particular token accordingly. If the offset is greater than 60 seconds, the server generates all TOTPs for the range $t+600\ s$ and $t-600\ s$. If a match is found, then it means the clock has really drifted or a person somehow acquired the TOTP for time t (*e.g.*, read the token display, termed *shoulder surfing*). For the sake of security, in such case, the server re-asks another TOTP to ensure the legitimate owner of the account has SecurID.

2.5.4.2 Gazzang zTrustee

The concept of *somebody you know* (discussed in subsection 2.4.2.4) has been, in fact, more or less adopted and modified in zTrustee [Gaz], an on-premises enterprise solution from Gazzang, a security company leaned for big data and cloud solutions. zTrustee introduces the concept of trustee, a party with a role similar to the helper in the *somebody you know* model [BJR+06]. A trustee can be a software or a human, and is able to approve or deny the deposit or retrieval of an object from the zTrustee server. A group of trustees may be required to vote for a single request. In this system, the trust relationship between clients and trustees is more abstract, and the functionality of the solution is not limited to authentication and can be

regarded as multi-factor authorization, since multiple entities vote for a single request. The solution can be integrated within the Linux Pluggable Authentication Module (PAM) stack for stronger authentication, and is suitable to supervise authentication of processes with nonhuman requests.

2.5.4.3 Authentify xFA

Authentify recently introduced xFA [Aut], a cloud-based passwordless MFA product that adheres to the usage of QR codes, voice biometrics and a PKI. The authentication process is as follows. The user accesses a target application, which creates an encrypted QR code for the user to scan using the xFA application, and then speaks a pattern to the device. The Authentify cloud stores the voice signatures and keeps a secure, encrypted connection with all devices using the xFA application as well as with the target cloud applications to which users want to authenticate. The xFA application encrypts and sends the voice signature and the QR code information to the Authentify cloud, which validates the data. Authentify then responds to the target application with the outcome, automatically logging in through the browser or denying access. Behind the scenes, a PKI ensures secure communications by encrypting every communication channel. Users key pairs are kept on the devices, while cloud applications are required to comply with the proprietary scheme in order for this to work. xFA is vulnerable to phishing if the target application keys are compromised and is also vulnerable to mobile malware. This solution resembles the main idea behind the project Google tried to implement back in 2012 [Nar12], though it was dropped.

2.5.4.4 YubiKey Hardware

The YubiKey hardware [Yub] of Yubico is also a good example to talk about regarding 2FA. Yubico is a recent company dedicated to make strong 2FA easy and affordable. Yubico is one of the members of both the FIDO alliance and the OATH organization. YubiKey hardware is available in various YubiKey flavors. All are small USB stick-alike and compliant devices to perform strong authentication on-the-fly. Various technologies are used to offer various features, including USB, Near Field Communication (NFC)[7], OTPs, and PKI. The devices can also be configured with a challenge-response protocol and standard static password. For example, the YubiKey Standard device can be used to generate TOTP codes for use in the Gmail 2FA mechanism as defined by the OATH.

[7]NFC is a growing low-range radio communication technology that enables exchanging data by joining two devices together or bringing them into close proximity. Applications include contactless transactions and data transfers, such as cryptographic material for authentication purposes.

2.5.4.5 Google Authenticator

Google Authenticator [Gooc] is an open-source 2FA software framework available for mobile and PAM-enabled platforms. The Google Authenticator application can be installed on Android, Apple iOS, and Blackberry devices. In this case, it can manage the HOTP or TOTP algorithms of various applications using 2FA by means of an offline token. Google Authenticator allows to scan QR codes with the seeding information required for any of the algorithms, easing the configuration of new entries and improving user experience. Alternatively, it allows to manually add each entry by entering the key encoded into a 16-bit base32 string along with the associated account information.

Google Authenticator is also available as a module for the PAM stack. It can be configured to ask for OTPs when issuing commands with root privileges or at the login screen. The software is compliant with the standard HOTPs and TOTPs code generators, but it can also be configured to use Message Digest 5 (MD5) in addition to the SHA variants previously mentioned, therein using HMAC with MD5. However, the Google Authenticator does not scale up well and securely on a multiple server setting [Rya12].

2.6 Conclusions

This chapter has elaborated on some basic concepts of cloud computing and authentication. The ideas herein discussed were mostly carried out while emphasizing the security topic. Basic concepts were introduced so as to better understand the chapter and the remaining part of this dissertation. The cloud computing technology certainly helps achieving the long-envisioned era of utility computing. Its crystal-clear benefits lower customer costs and potentially increase their business productivity. However, as shown in subsection 2.2.4, the research on this field has been greatly focused on the security side until now, despite the proliferation and success cloud computing is making throughout industries.

Authentication is no longer restricted to the application logic. It is increasingly becoming more attached to hardware devices, since people are more accustomed and dependent on them for the daily life. Combined with standard password login, MFA brings device- and user-centric authentication a step closer. The rise of SaaS, PaaS, and IaaS clouds magnified the problem of password management. Quickly enough, industry vendors sorted out the SSO approach. Centralizing authentication to some IdP is like putting login credentials in an outsourced vault, accessible from anywhere and everywhere, provided an Internet connection is given. Moreover, two preponderant aspects caused 2FA to be adopted in recent years: the necessity to move beyond static password and the revolution of the mobile era. As such, the research on the literature is focused on mixing independent concepts and technology to create unified and strong authentication mechanisms. The integration of smartphones with MFA schemes is rising. For

now, QR codes are being widely used for creating various applications for authentication. It is expected to see NFC technology to become preponderant for both offline and online authentication in the future.

Most authentication mechanisms reviewed in this chapter are applicable to cloud computing management interfaces, and to both SaaS and PaaS applications as well. However, after the authentication phase, they remain vulnerable to the myriad of issues a progressive and long web session based on cookies is. The next chapter overviews the main security issues impacting the entire process of authentication, but first focusing on enumerating security issues in cloud environments. The model for authentication on cloud environments presented on chapter 4 elaborates on the possibility of adding layers for authentication in separate VMs while keeping the legacy mechanism. This way of thinking was motivated by the study presented in this chapter, which clearly showed that MFA is the immediate solution to the authentication problem, along with the usage of new technology to decrease the burden that such procedures impose to the users.

Chapter 3

Security in Cloud Computing and Authentication

This chapter enumerates several security issues arising from, or associated with, the cloud computing model. The discussion then evolves to the authentication topic, describing the issues around classical and emerging approaches, introduced in the previous chapter. The chapter provides a general picture over the security state of these topics.

3.1 Introduction

Since the cloud computing business model implies Internet communications, along with online authentication, the most obvious security issues arise from the cyberspace. Nowadays, the Internet is utilized by users of all over the world, each of which with its own purposes. Throughout the years, a dark community started to step up and take shape, while spreading the word of cybercriminality. In recent years, some of the individuals of that community have been focusing their efforts on specific targets, becoming, in most cases, more specialized and dangerous. Overall, the cyberspace of the Internet is a very entropic environment with regard to security and privacy. Since the Internet and how it is utilized is mostly relevant for businesses conducted therein, and particularly for the cloud management interfaces, Section 3.2 superficially addresses the topic of the cybersecurity. The contents therein discussed are more detailed in the fourth scientific publication [FSG+13a] of this masters program.

The previous chapter described the cloud computing model, introducing the basic concepts from the security perspective, whenever possible. It was said that cloud computing offers several benefits from the customer point of view, but that it also creates new challenges from the security standpoint. Sharing an infrastructure that is accessed by multiple users and in which sensitive data or computations may be carried out, creates a trust barrier. In addition, the underlying cloud technology also poses severe issues that may jeopardize the cloud operation on any of the service delivery models. The brunt of the security issues impacting cloud computing environments are discussed in Section 3.3. That section is mostly based on the security discussions of the first [SFG+14] and second [FSG+13b] scientific publications of this dissertation.

The previous chapter also addressed the authentication topic. It clarified how authentication is typically implemented using classical approaches, on web-based communications, both in the initial verification phase and in the long-term. Novel authentication trends were later described, followed by the state-of-the-art in the field. Authentication trends show a move to-

ward the convergence of protocols, unifying various technologies while making authentication more coupled with devices and people. The motivation for this is clearly the degradation of the classical one-factor login. Elaborating on this topic further, Section 3.4 explains the main factors for such degradation, further highlighting the necessity for this transition. Next, Section 3.5 points out the challenges of new authentication trends and mechanisms. The ideas discussed in these topics are partially based on the second [FSG+13b] and third [SFFI13] scientific publications of this dissertation. Section 3.6 presents some noteworthy real incident cases related with the subject of this dissertation.

3.2 The Internet and the Cyberspace

With the degradation of the cyberspace with respect to security, the cybersecurity field has been gaining popularity in recent years thanks to surges of malware, spam, phishing, vulnerabilities, and other threats of the like. Behind the curtain, these threats have been transforming part of the cyberspace into a deep underground web [PA12]. The threats come from hacktivists, cybercriminals, and Advanced Persistent Threats (APTs) or nation-states [SE13]. These are the three main profiles classifying individuals or groups conducting malicious activities on the Internet. They are turning the cyberspace into an open battleground. Hacktivists are motivated by political, religious, or patriotic reasons, protesting against wrongful laws and corrupt or repressive governments. Cybercriminals aim at profiting without looking at means, while APTs may be hired for conducting specialized hacking, including governments hiring those groups for carrying out cyberespionage to other countries.

Interestingly, the XaaS term has been adopted by the cybercriminal underground, delivering services of the dark-side similarly to cloud computing, though it does not relate with cloud technology. This is the case of Cybercrime-as-a-Service (CaaS) [Pag13], a broad term referring to any type of cyberattack provisioned in a pay-per-use manner. Examples include the Distributed Denial of Service-as-a-Service (DDoSaaS) [Ker13b]. Along with these trends, malware writers are adapting, shifting their efforts to new computing platforms, namely cloud computing and smartphones. Spam and phishing schemes evolve accordingly to overcome security measures and take advantage of casual, but very particular events (*e.g.*, pope election) from the real world to trick people with social engineering techniques. It all comes down to the awareness people have at the moment of interpreting an email and click a link or opening a file [Tho13]. The impact of some of these threats in cloud computing and authentication is better discussed throughout this chapter in proper places.

More recently, the world has come to know [Sch13a] of the Internet-wide mass digital surveillance programs of the National Security Agency (NSA) and of other agencies from the United Kingdom. Thanks to Edward Snowden, an NSA ex-employee, there are evidences of the NSA installing taps at fiber optics cables, forcing subpoenas to place backdoors on major service

providers like Google and Facebook and ISPs, or even intentionally weakening cryptographic standards. The apparent motive behind such a surveillance is to fight off terrorism by detecting communications of key people involved in such foul acts. But the NSA is collecting data from everyone, possibly monitoring millions of records daily. This is highly problematic from a privacy point of view for both end users and companies which, associated with the cybercriminality and cyberespionage, makes the Internet cyberspace a harsh place.

So, the Internet and the cyberspace is nowadays used for disputes between individuals, cybercriminal groups, and governments, harnessing the power of Internet communications to exfiltrate data and extract information or to steal money and intellectual property. There is an ongoing cyberwar and surveillance that affect Internet users and corporate businesses as a whole. This general perspective of the cybersecurity state is relevant for both cloud computing services and authentication, not to mention every other online service.

3.3 Cloud Computing Security Issues

The cloud computing paradigm came to agitate the IT world in two ways. The first way was positive, since it generated lots of discussions around the benefits of the technology, which led to its mass adoption throughout the world. On the other hand, several security issues emerged. But on the good side, the awareness quickly spread and now commotions on the academia and on the industry lean toward addressing its issues in the first place, before exploring other parts of the technology. Below are included discussions about cloud security issues, paying attention to the management interface problem.

3.3.1 Virtualization

Cloud computing thrived mainly due to the proliferation of virtualization technologies, which only became possible when Intel and AMD added full virtualization support for their CPUs in the mid-2000s. Hardware used to be by default designed and built for a single OS, and thus it was not possible to run concurrent OSes in an isolated manner without sharing CPU registers and caches. However, IaaS clouds brought security issues intrinsic to the virtualization process. VMs can be instantiated on particular machines at some instant in time. But, later on, VMs can be migrated from one place to another because of cloud balancing. This implies a loss of control [XX13] in the instances, meaning that it is impossible to pin-point the exact physical and logical location (*e.g.*, switch network ports) of the instances. Moreover, VM images are usually large-sized files, and it is thus hard to employ common cryptographic techniques without relinquishing computing overhead [VRMM11]. Because IaaS clouds usually handle a large number of VMs, the management of the repositories of VM images may be quite complex [WZA+09], along with the migration of VMs that may be susceptible to MitM attacks [OCJ08].

Running various OSes over the same machine greatly facilitates configuration and management.

But the co-resident setting is not yet perfect. Research on this field is notably focused on such IaaS layouts. For all purposes, running co-resident VMs means that the same physical CPU core cannot be used at the same time by different instances. Researchers have exploited this scenario by trying to understand the computational usage of doing specific computational tasks in specific co-resident VMs, greatly focusing on the extraction of cryptographic material. Such attacks are popularly known as side-channels and covert-channels [RTSS09, BNP+11, ZJRR12]. Others have explored [XXHW13] the memory deduplication technique, which allows to merge identical memory contents from different VMs to improve memory efficiency, while others were able to extract [RC11] cryptographic material from memory dumps, provided an insider position was given. Random number generation many times relies on entropic events gathered from noise sources. On computers, such noise sources are hardware peripherals. However, because of the virtualization layer, the Linux kernel may not gather sufficient entropic events to output random material with the same efficiency [FSFI13], when compared to hosts. This may be critical for VMs generating cryptographic material. One other problem relates with malware. VMs are useful for forensic analysis since one can install a malware sample on an isolated VM and monitor its behavior without jeopardizing the security of the host and the local network. However, malware has evolved, becoming aware [Ort12] of the environment and being capable of detecting the presence of VMs.

3.3.2 Storage and Computing

It was previously said that VMs may not be physically located within a IaaS computing environment. The same holds true for other service delivery models in the sense that one loses control over data as well, since data is outsourced to some cloud or service provider. For redundancy reasons, data may be backed up to different locations. Large cloud providers owning various data centers may achieve this by copying the same data to a different facility. If one data center becomes unavailable, the other steps in. Not only this renders loss of control over the data location within the data center, but it also raises a multi-location issue [ZZX+10], for which implications are further explained in the next subsection. The fact that the underlying computing hardware is shared may also raise problems concerning availability. A bogus VM or some service hosted on the cloud may be purposely consuming resources in order to deny resource allocation to other legitimate services. There is also the possibility of dishonest computing by the physical servers, administrative errors in backups, restores or migrations [XX13]. For PaaS clouds, it is important to assure safe threat termination and isolation, wiping remnants of processes like memory objects and references to avoid leaks. Neither .NET and Java platforms fulfill [RMVC+12] these requisites for shared and multi-tenant PaaS clouds. Some providers (*e.g.*, Google) physically wreck hard drives that are no longer required and may be thrown away. Another issue is raised if such destruction policies are not correctly implemented, along with the data withhold by the hard drives [AFG+10, BW12].

3.3.3 Trust, Legality, and Compliance

Adopting cloud solutions requires depositing great amounts of trust in the cloud provider, especially for enterprises. The continuity of enterprises relies on business secrets. Storing such confidential information on IT systems belonging to a third-party contractor like a cloud provider is risky from a business viewpoint. In addition, it is hard or impossible to tell if cloud providers strictly comply with the service terms specified in the SLAs[1], including security and privacy matters, due to the NSA surveillance case for instance. Yet, one may never be sure that cloud providers are not dealing under the table with such agencies and violating SLAs without customer knowledge [ENI09, XX13]. In addition, it is not clear if cloud providers use subcontractors or resort to other means for selling their businesses [Pea13].

It was said in the previous subsection that data may be copied to different data centers for backup purposes. However, large cloud providers may own data centers spread across different countries, wherein different laws apply to digital data. This raises several issues in the legal domain regarding data flowing over borders and jurisdictional interception for data stored overseas. The problem is that outdated laws and acts [ZZX+10] are not applicable to his new computing scenario created by clouds and their interconnection. In fact, interoperable clouds speaking the same protocols are required for defining interchangeable rules in order to make data migration a reality, and thus free costumers from the vendor lock-in issues [ENI09, AFG+10, GMR+11]. Because clouds exploded rapidly, cloud providers started developing their own protocols. As a consequence, the vendor lock-in issue arose, inhibiting customers from migrating their services to other cloud providers. Overcoming this issue calls for open standards.

3.3.4 The Problem of the Management Interfaces

Clouds can be one of three possible types. SaaS models provide specific pre-packaged software applications written by the cloud or service provider. PaaS cloud models supply the means to deploy applications on the cloud written by the customers, following a Service-Oriented Architecture (SOA). Both of SaaS and PaaS clouds are typically built using standard web programming languages and related technologies. The difference in PaaS clouds is that they may run applications on a multi-tenant and shared infrastructure. As such, web applications may be prone to the traditional vulnerabilities that websites and other online resources are, namely of Structured Query Language Injection (SQLi) and Cross-Site Scripting (XSS). Moreover, data travels an umpteen number of network points before reaching the cloud, and thus MitMs attacks are possible. The protocols used for communications and web services are restricted to what exists in the Internet, and the developed application typically mix HTTP and HTTPS for content delivery [Tro13], and suffer from flaws in the Simple Object Access Protocol (SOAP) [JGH09]. As such, vulnerabilities or shortcomings in these protocols or related technologies are crucial for

[1] SLAs specify how services are furnished to customers, detailing the general usage of the service and other conditions. Security implications should be included.

the security on the aforementioned models.

Management interfaces of IaaS clouds, on the other hand, may prove to be more risky. These interfaces are used to orchestrate a virtual data center, including the creation, destruction and management of VM instances, the grouping of VMs and construction of virtual networks, as well as security properties. Those interfaces may contain information about the cryptographic keys used to access each particular VM. But, just the fact that VMs can run crucial services for the customers should alone be regarded as being of utmost importance, because if an attacker make of a compromised account a stealthy attack base, he or she may copy, migrate, tamper or terminate the instances, resulting in losses to the customer. There is also the problem of availability. Clouds are well regarded in this matter, usually achieving high and reliable uptime. Nonetheless, just a small fraction of downtime may prove costly for specific high-value businesses; and one thing is having the control in the customer side, while the other is passing it onto third-party providers. For example, Amazon EC2 was subverted by spammers in 2009, which caused major service disruptions [CPK10].

On traditional network engineering, management interfaces with administrator capabilities are deeply within the enterprise internal network, accessible only from the private addressing space of the network, and behind several security controls, such as firewalls. Some companies like big ISPs may even allocate their addressing space specifically for frontend, backend, backup, and management tasks, delineating particular IP zoning areas. The frontend perimeter gets different address assignments than management addressing, for instance. On cloud computing, this is clearly and largely overstepped. The management interfaces and the subscribed services are accessible from the Internet, and get assigned public addresses or unique domain names.

Section 3.2 reflected the cybersecurity concept as a rising trend due to the maliciousness found on the Internet and the cyberspace. In fact, the main issue of the management interfaces of cloud computing concerns the cybersecurity state. Clouds can be characterized by their openness and dynamics to the outside, relying on the connectivity to generate traffic loads between the clouds and the customers end. The network perimeter of clouds is very thin, and the only thing in between an adversary and a target resource is authentication, independently of the origin of the connection. Attackers may conduct malicious activities without raising high suspicion. If the cloud provider does not conduct monitoring activities, attackers are free to execute intelligence gathering [SE13] by scanning and roaming the cloud network (to some extent), and particularly by exploring the management interfaces. Attackers can subscribe fake services in order to learn the internal workings of the cloud. In PaaS clouds, customers may develop whatever they best see fit in their applications with regard to security, but they still remain susceptible to the outside. For example, the Plesk Panel interface was recently found vulnerable [Rep13]. The Plesk Panel is used for pumping up hosted sites, for which many are on clouds. The vulnerability allowed to execute arbitrary code, which was hastily exploited by a

botnet. It was nonetheless uncovered and shutdown quickly as well. This upfront exposure of frontend interfaces for critical enterprise services proves to be risky enough to reconsider the commitment in adopting cloud solutions.

3.4 Passwords and One-Factor Login

Passwords have always been perceived as one-factor login systems. Coupled to a unique username in the system, an identity is successfully identified by proving knowing both the username and the associated password. However, the countdown for the end of password-based authentication might already have started with the emergence of the FIDO alliance and the OATH organization. The development of stronger and newer authentication methods based on emerging technologies is ongoing. Such effort is best illustrated by explaining why the password is now considered unsafe as a one-factor login approach.

3.4.1 Bad Practices and Awareness

Passwords are a subject of increasing study from both the academia and industry vendors, but also from the cybercriminal world. In part, the source of the password problem is related with some of the practices employed for password-based authentication, along with the awareness on this topic. Some people find it difficult to memorize passwords in both professional and personal environments. As such, it is common to see insecure low-sized passwords with trivial character combinations that are easy to guess. Actually, some even might choose stronger passwords. But in such case, they write them down on a piece of paper, putting it under the keyboard, inside a drawer, or attached to the monitor, despite enterprises defining global security policies and identifying guidelines to avoid these risks. Moreover, employees may share their passwords with a coworker, a friend, or even a friend of a coworker, even after receiving specific training on password security [Tow09]. This lack of security awareness combined with bad practices may have serious consequences. The human factor may very well be considered one of the weakest links in any computer system, no matter what kind of technology is employed. Security cannot be resolved by technical solutions alone. Security controls must adapt accordingly, but the mentality, awareness, and wisdom of the users and employees has to be changed as well. This principle has been for decades a major barrier in computer security. If overcome, it would greatly minimize the impact of spam, phishing and malware.

Several reports [Imp10, Doe12, Tru13] point out these issues with thorough analyzes. Scary password habits are therein revealed, showing trivial, low-sized words as the most common passwords, containing no special characters or numbers. For example, it is common to see the first letters in capital form or a name followed by a year (*e.g.*, “JaneDoe2013”). Other popular combinations include replacing letters or logical words with similar numbers. For example, “Jane” could become “J4n3”. Good enough mixtures of lowercase and uppercase characters scrambled with numbers and non-alphanumeric characters should be rather used so as to as-

semble a good amount of entropy for stronger passwords. Alternatively, logical phrases (*e.g.*, "Today is a sunny day!") also comprise strong passwords that achieve that same aforementioned security properties, but remove most of the memorization burden. In addition, the use of Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) is on a downhill [Tay]. Several online services provide a CAPTCHA in addition to the standard username and password combination in order to distinguish bots (*e.g.*, spammer malware) from persons. However, advances in image analyzes to recognize each character in a CAPTCHA image is outpacing this approach, rendering it a weak method.

Despite the previous issues, the Ponemon Institute surveyed [Pon13] user attitudes in the light of the deluge of web services with online authentication, finding good results. Almost two thousand individuals of several ages were polled. The study found that the majority did not trust systems whose only authentication method is based on archaic passwords. Moreover, the majority also prefers a multi-purpose identity credentials, meaning that an IdP would provide a single, centralized sign-in location for third-party cloud-based services, such as homebanking and online shopping. The surveyed users also manifested their inclination toward biometric-based authentication, being the preferred method voice recognition, Radio-Frequency IDentification (RFID) chips, and mobile devices—good signs for MFA.

3.4.2 Password Breaches

Another source of the password related problems is associated with the massive breaches of passwords that have occurred throughout the years. This may be the most important factor contributing to the degradation of password safety. Password breaches, like the case of the LivingSocial hack, which exposed fifty million salted and hashed passwords [Liv13], provide the basic elements necessary for finding construction patterns in password sets and, consequently, insights into password cracking. Epic data breaches amount to over 100 million records leaked [Inf]. Adobe has been the most recent victim [Ker13a] of data breach, leaking the source code of company products, but also records of usernames, passwords, and encrypted credit card information belonging to 2.9 million customers. A data breach is a likely risk that can surface from poor security, accidental or unintentional leaks, lost media, insider threats or from vulnerable frontend systems. For example, SQLi is a very popular attack vector since it can access backend databases by exploiting some unattended sanitization hole.

3.4.3 The Exponential Wall of Brute-Force

Increasing the password length and character space has always been the greatest handicap to brute-force attacks. In the thirties and forties, the best example for this is the cipher machine Enigma from the German military. A cascade effect resembling Pseudo-Random Number Generators (PRNGs) would take a seed code to produce the same ciphertext for the same plaintext, and vice-versa. Eventually, the machine code was broken due to a flaw. From that start-

ing point, cryptographers brute-forced several combinations with basis on that flaw, but they needed to do it combining mechanical and electrical means which would take, at the time, long times to process. The cracking procedure was based on an elimination setting, meaning that the cracking machine named Bombe would assume certain combinations and would try them sequentially. Today, the process of cracking is not much different.

Authenticating on one-factor login systems may be achieved by simply inputting all character combinations to the web page, if no limit is imposed. Nonetheless, brute-forcing is usually done offline to crack the millions of passwords obtained from the data breaches over the years. Such passwords are usually hashed or salted and hashed, making them more resilient against cracking. However, such high volumes of real-world data allowed crackers to compile huge rainbow tables and dictionary lists [57u13, Daz] containing hundreds of millions of passwords. Attackers can refine their models for selecting user passwords and, therefore, design faster password cracking algorithms [KKM+12]. Instead of following an elimination process like the Bombe cracking machine, some algorithms choose candidate passwords based on statistical analyzes. Never before building password cracking lists and carrying out brute-force attacks has been so efficient and easy.

In addition to all of the aforementioned facts, computers nowadays pack great processing power in Graphics Processing Units (GPUs) and multicore CPUs. Dedicated custom-built hardware is even more problematic in this case, as it has been recently shown [Tra12]. Researcher Jeremy Gosney built a cluster with 25 AMD Radeon GPUs capable of shredding password hashes with a throughput of 348 billion hashes per second. A relatively large panoply of powerful cracking tools is also available, including John the Ripper [Ope], Aircrack-ng [Air], Hashcat [Hah], Ophcrack [Oph], Extreme GPU Bruteforcer [Ins], and CloudCracker [Tho].

Despite of what was said, brute-forcing a very long password (over 40-50 characters) is nonetheless unfeasible, and access to specialized cracking hardware is typically only possible to a handful of privileged persons, who built computing platforms for that purpose, or for big agencies like the NSA, which may have large budgets allocated to such projects and a significant amount of computing resources. Increasing the password length character by character exponentially increases the cracking time, eventually hitting the so-called exponential wall of brute-force cracking [Goo12]. Nevertheless, very long passwords are more difficult to remember or use on a daily basis. By taking all of the aforementioned ideas into consideration, it may be concluded that the exponential wall of brute-force cracking is not normally an obstacle, since the specialized algorithms do not hit the wall before cracking is successful.

3.5 Challenges in Authentication Trends

Heretofore, authentication using passwords comprises the most implemented access method in the world, hence the discussion dedicated to this subject alone. Nonetheless, new trends are emerging all over the industry and academia, mostly motivated by the objective of improving the security state of authentication in general, including both offline and online processes. When no time to mature is given to these new approaches, they are often synonym of new security issues, as demonstrated by cloud computing. Naturally, novel authentication approaches have been scrutinized for security in the last few years. This section looks into some of those issues.

3.5.1 Multi-Factor Authentication Security

In 2005, Bruce Schneier predicted [Sch05] that the early adoption of 2FA would result in a significant drop in fraud but, in the end, only a negligible decrease would be observable. Today, 2FA is considered a preventive solution against account compromise. The *something you have* adds an extra security layer for cybercriminals to overcome. Four vectors are, nevertheless, laid out as options for attack against 2FA. The more obvious one is theft of the physical device that generates OTPs, if that is the second factor being used. The second is the so-called shoulder surfing, which consists in visualizing the display of the token device without being noticed and use the observed values to authenticate. The remaining two are MitM and malware. A MitM attack can be mounted by building a phishing website that, at a first glance, is visually similar to the legitimate website. The page is, however, hosted in a malicious domain, and captures user credentials and OTPs to perform any kind of operation on the victim service, such as money transactions. On the other hand, local malware, such as MitB, can patiently wait for users to login and capture `GET` and `POST` requests, relying on them to receive or generate OTPs on their mobile devices and enter those codes in the web applications. Mobile malware is also an issue in the case of 2FA based on mobile devices. Furthermore, 2FA via SMS messages suffers from delivery delays and of signal coverage spots (*e.g.*, basements). The security improvement of 2FA can, nevertheless, reduce the window of opportunity for fraudulent campaigns and narrow down their scope. Stealing an OTP code is only useful for a short time frame and for a single use, proving to be harder and more costly for adversaries.

OTPs efficiently solve many security issues thanks to the dynamic nature of OTP. The security of the HOTP algorithm relies on the distinct inputs of counters to generate uniformly and independently distributed four byte dynamic binary code strings. Brute-force attacks are hardly an option, since backend servers can simply re-ask more OTPs when failed insertions are detected. For six digit codes, guessing an OTP at first has a probability of $\left(10^6 - n\right)^{-1}$, where n is the number of past observations. Therefore, knowledge of past OTP observations almost does not help. The TOTP security is, consequently, dependent on the properties provided by HOTP. Both algorithms can be used for a variety of network applications, including remote VPN

access, transaction-oriented web applications, and as a supplement to traditional OSes login and standard static password authentication. Their security is directly related to the security of the cryptographic key used which is used as a seed for the OTP generators. Moreover, while hardware tokens cannot be copied, software tokens may be subject of duplication.

It was said in subsection 2.4.2 that MFA may mix two or more approaches of total amount of four: the *something you know*, *have*, *are*, or *do*. The latest two approaches are, however, somewhat cloudy when discussed from the two following perspectives. First, a program cannot simulate a biometric feature, and second, the pieces of information used in such approaches are, for all practical purposes, unchangeable and not volatile. This means that, if biometric data gets compromised, it is very difficult, if not impossible, to change the fingerprint or the retina, whereas passwords and keys can be changed on-the-fly with little cost. Moreover, fingerprints are virtually in everything a person touches, which can be forensically copied. Another problem with fingerprints is that not all of its features are unique [YJIZ12], and thus it is important to select the best ones. Regarding the *something you do*, e.g., speaking patterns can be easily recorded. These shortcomings point out possible weaknesses in these authentication factors.

Regarding the SSO approach, there is also one challenge. A single compromised account can give access to all the online applications compliant with that particular IdP. As such, attackers can leverage this centralization of the authentication procedure to thoroughly impersonate someone.

The authorization domain also raises particular concerns regarding malicious third-party applications that may endanger the welfare of popular platforms. This is the case of Facebook, which is one of the most targeted social platforms by malware since it centralizes several third-party applications. Malware may spread [Sch13c] through the social network with profit intents by being released on such applications, that are personally authorized by each user. Facebook actually has automated processes for cleansing malicious applications, but when something goes wrong there might be an outage. This happened recently with Facebook [Zar]. For shared cloud environments, there is always the possibility of another co-resident customer to host a service that integrates various sources of data and applications, thereby motivating this problem.

Reporter Mat Honan had all his devices wiped out [Hon12] after being victim of hacking. The hack began through social engineering on Amazon and Apple support structures. 2FA could have prevented this incident. Since then, Apple has added 2FA to Apple ID [Fri13], following what Amazon Web Services (AWS), PayPal, Dropbox, Facebook and Google [Fon13] did. Twitter and LinkedIn have also done the same. Evernote got breached in their network, potentially leaking usernames, emails and encrypted passwords [Dav13]. For security purposes, it was decided to reset the passwords of nearly 50 million accounts. Quickly after that, a plan was rolled out to implement 2FA in Evernote SaaS system. For all these reasons, it is better to have MFA than not

having it.

3.5.2 Flaws in Mobile Devices

Placing authentication responsibilities on mobile devices can be problematic, though they undoubtedly improve the user experience. It is up to the vendors to ensure that screen unlocking is secure, may that be through standard passwords or biometric scanners, like the face unlock supported in Google Android devices. NFC can also be used to detect in the near proximity some personal item a user carries. Regarding the standard password unlock, bugs in both Android [Mim13] and Apple iOS version 6 [Sli13] and 7 [Gre13] have allowed bypassing the unlock screen, though with some potential limited functionality, but sill pointing out issues in lock safety. Apple now ships their latest iPhone model with TouchID, a fingerprint scanner to unlock the device and confirm high-value operations. However, the technology was broke [Cha13] by simply making brushed up printed version of a previously photographed fingerprint on a glass surface. This is definitely concerning for the security state of biometric approaches for authentication, though it depends on the scanner.

3.5.3 Quick Response Codes Security Issues

QR codes were reviewed in the previous chapter as an alternative technological means to exchange data for authentication purposes, easing the procedure for the user and providing a more enjoyable experience. But, QR codes can only encode a limited amount of data. They are commonly used to encode a website URL. This is a security issue. Malicious QR codes can encode shortened URLs that obscure destination links used to serve malware, like unwanted applications or mobile viruses, or even phishing websites. Only QR codes trusted from known sources should be scanned by humans to overcome this issue, which requires awareness again. In terms of automated reading processes, it should be mentioned that it may be possible to encode some data for injecting commands or tricking the system to perform some fraudulent operations [KLM+10]. These attack vectors must be taken into account when devising solutions using this technology.

3.5.4 Malware

Malware is one of the driving forces steering cybersecurity. For authentication, it is a headache. In the context of the current password prevalence landscape, the main goal of malware is to steal usernames and passwords. Due to the underlying profit, cybercriminals mainly build malicious code to steal homebanking credentials, but email accounts are targeted a lot as well. A compromised email account, if configured as a recovery email address, can be used to try and get into other online services. McAfee [McA13b] reports a steady increase in new password stealer samples since the first quarter of 2010. For example, Sellmer, from Microsoft Malware Protection Center, has described the behavior [Sel13] of a new variant of the Reventon ransomware that downloads a password stealer Dynamic-Link Library (DLL) from its Command-and-

Control (CnC) servers. More concerning, it was recently reported by FireEye in its Advanced Threat Report of the second half of 2012 [Fir13] that malware events occur, in average, once every three minutes at a single organization. Moreover, 50% of malware downloads additional malicious binaries within the first minute of infection, according to Websense in its 2013 Threat Report [Web13].

Due to the explosion of the smartphones era, mobile malware has also been rising [Web13]. Mobile malware is cresting as never seen before, with new malware being detected at breakneck speed. It is case to say, *cybercriminals go where users go*. Focusing Android mobile devices, a 2577% increase was registered in the 2012, as reported in the 2013 Cisco Annual Security Report [Cis13]. However, such high percentage only contributes to a 0.42% mobile malware slice out of the whole set of top web malware threats for 2012, but reluctantly show malware writers adapting to IT trends. These statistics and progression in cybercriminality raise alarming concerns regarding password-based authentication and MFA mechanisms relying on the mobile device. For example, the Pincer 2 trojan for Android devices is capable of intercepting inbound SMS messages and forward them on [Doc13]. Indeed, this capability can thereby compromise 2FA systems that rely on SMS to deliver OTPs on mobile devices. The trojan, which is maliciously installed via a fake certificate, awaits a set of instruction commands set by CnCs that affect its behavior, being the instruction `start_sms_forwarding` of particular interest. This directive tells the malware to listen for SMS messages received from a phone number appended to the command, ruling out unwanted messages received from unfamiliar numbers. Maor [Mao13] also described a MitB malware variant named Ramnit. Ramnit uses a 2FA scam to trick homebanking users unknowingly wire transferring money to mule accounts. Both these examples illustrate the advent of malware aware of 2FA mechanisms.

3.6 Real World Incidents

The academic research side and industry reports show interesting insights into security issues by means of theoretical and empirical analyzes. Assessing the practical impact of some problems may be hard enough since many works are Proof-of-Concepts (PoCs), while other studies may require to get the hands on a fully operational system to conduct real tests, which may be tough or not be possible to build at all. In order to get an idea of the practical effects of some security issues discussed throughout this chapter, this section presents below a few real world incidents. They illustrate the possible impact on the society and economy, and even on the common individual, therefore highlighting the importance to address such security issues.

3.6.1 The Eurograbber Trojan

The notorious Eurograbber trojan gained wide popularity when it was discovered [Ver12] to have stolen an estimated amount of 36 million euros. Such a high and remarkable profitable attack was possible because the malware targeted specific banks within the euro zone. The malware

was a customized version of the popular Zeus malware kit, for which the corresponding source code leaked in 2011. The attack initiated with a spam email containing a malicious link to download the trojan into the computer. The trojan would then remain dormant until it detected users authenticating to the targeted banks, injecting JavaScript code into the browser, tricking the user that the bank was upgrading some security measure. In the meantime, it asked to insert the mobile phone so as to send a bogus SMS message with another malicious link pointing toward another trojan, specific to the mobile device. The purpose of the mobile trojan was to specifically circumvent 2FA employed by the banks to confirm transactions. From there onward, users money would be unknowingly transferred to mule accounts[2].

3.6.2 RSA SecurID

The RSA SecurID product is quite secure and it is regarded as a good second authentication layer. Hardware tokens are tamper-resistant, meaning that they cannot be duplicated. The main attack point may very well be the backend servers that also keep clocks ticking according to the TOTP algorithm. For the TOTP algorithm to work, both backend servers and the user tokens must run with the same seed parameters, as described in sub-subsection 2.4.3.1, namely the secret key. But those backend servers must be somehow linked to the authentication system on the frontend. The RSA was hacked [11] back in 2011, leaking password protected files. Official statements suggested that the SecurID devices had to be replaced only to reinforce authentication, but not the specific details of this hack. This points to a relation of the attack with the 2FA SecurID product, but also to a vulnerable point of the 2FA systems in general. If the seeding keys would have been stolen, all of the SecurID hardware tokens would be useless and provide no extra security. In such case, it would be necessary to reseed or replace all of the tokens conveniently.

3.6.3 Twitter Two-Factor Authentication

Twitter recently rolled out 2FA in the face of the adversity and number of cyberattacks. Several Twitter accounts belonging to companies or organizations have become highly attacked targets by hacking and hacktivism groups, namely the Syrian Electronic Army, which try to use Twitter as a means of protesting. Such targets include the British Broadcasting Corporation, Reuters, Al Jazeera, CBS News, and Associated Press. The incident related with the latest target caused millions of dollars in losses in the financial market, and announcements [Ass13] indicate that phishing was the culprit.

The 2FA authentication, however, was not as secure as expected [Hyp13]. 2FA is in general phishable, meaning that a forged page can be setup to capture the first and second factors of

[2]Mule accounts belong to money mules. Money mules are persons who are fooled to transfer illegally acquired money coming into their accounts to other bank accounts, typically in other countries. Money mules are usually recruited on online employment sites, by tricking them into thinking that those are legitimate jobs, since a small rate is earned for the service.

authentication. Additionally, local malware can also achieve that purpose. Twitter, however, complicated the issue because it has a feature that enables to tweet via SMS. This feature requires a mobile phone to be associated with an account and, for roaming reasons, it can be deactivated by messaging Twitter with the word `STOP`. However, besides stopping the SMS tweeting, the message also removed the mobile phone number configured on the accounts. If 2FA was enabled, the `STOP` message in turn disabled 2FA. Other entities but the legitimate owners could use SMS spoofing to disable the 2FA system of Twitter, in turn locking out victim accounts. More worrisome, if a malicious entity got hold of a compromised account somehow, it was possible to activate 2FA with a spoofed phone number. Twitter did not validate the phone number by sending a confirmation SMS and, even if a password reset was performed, the system would then send the 2FA code via SMS to a number not belonging to the legitimate owners. Since then, Twitter has improved [Jim13] their 2FA procedure and is now based on public-key cryptography without relying on SMS messages, but on a mobile application.

3.6.4 The Dropbox Client

Dropbox is a popular cloud storage service. Typically, these services offer computer and mobile applications to synchronize files across various devices in a centralized manner. That is, devices upload files to the cloud, which is responsible for distributing the data throughout the remaining devices connected to a particular account. Naturally, the devices are required to be linked to that account by carrying out normal authentication through the applications. However, if 2FA is activated on the Dropbox website, the application does not ask for the OTP as the second factor. This is the first shortcoming of the client software.

Although the Dropbox client software is hard to reverse engineer because of robust bytecode obfuscation and encryption, researchers Kholia and Węgrzyn [KW13] were able to learn its internal workings and API for communicating with the online services. They also found how to hijack Dropbox accounts, being able to snoop data before being encrypted on the sender side and after being decrypted on the receiver end. Also important, they discovered how to bypass Dropbox 2FA system that, as it turns out, is only active on the Dropbox website. The Dropbox client allows to automatically open the website without requiring authentication. This feature is done through a unique `host_id` value that is kept encrypted on a local SQLite database. The Dropbox security ecosystem relies on this `host_id`, which is possible to obtain through code injection techniques. The Dropbox client uses it to generate auto-login URLs not requiring 2FA, thus being possible to access all data of the Dropbox account associated with the `host_id`, regardless of the password. Malware developments may be seen in the future for exploiting this vulnerability if a patch is not issued.

Although reverse engineering the Dropbox client is not a direct problem of neither authentication nor cloud, it shows how deficient implementations of software can impact crucial

cloud-based systems and authentication measures. MFA mechanisms should be revised throughout all the application development workflow, reviewing all the features and functionalities, so as to make the security layers consistent with the specifications and implementations.

3.7 Conclusions

This chapter has overviewed the security state of both cloud computing and authentication topics, shedding light over their main issues. Clearly, the cloud computing model is highly affected by several security issues spanning throughout the cloud service delivery models, orthogonally to the cloud deployment models, and its business and operational models. As such, a good enough amount of forethought should be placed on clouds before committing to a specific cloud service delivery model, in replacement of traditional IT systems. The management interface problem is mainly affected by the issues inherited from the Internet. The openness of cloud environments diminishes the security and increases the number of attacking vectors and, from the costumer point of view, that is something important to consider. In the second half of the chapter, the subject is discussed using real examples of problems related with authentication in the last few years, so as to better transmit the way it should be implemented to avoid those problems, which is one of the main objectives of this work. 2FA seems to be the short term answer to the password-based authentication problems, mostly because of backward compatibility. Nonetheless, there are better ways of achieving the same purpose on applications devised from the ground up which should be considered in that case, namely based on public-key cryptography and ZKPs. The next chapter elaborates on a model that enables cloud providers to incrementally increase the security and implement better authentication techniques resorting to virtualized resources.

One of the main reasons for the need of new authentication approaches is the human factor. The struggle to spread the awareness on security and utilize strong passwords is not new. But people prefer easier methods allied to good user experience while carrying out secure authentication, replacing one-factor login systems based on the combination of usernames and passwords. To answer these requests, the industry is carving novel protocols and mechanisms that will unite the benefits of independent methods. It is also noticeable from the discussions in subsection 3.5.1 that service providers only realize that stronger authentication should be deployed when a security incident takes place. This passive way of thinking is wrong. It should be the other way around; security should be regarded using a top-down approach, proactively implementing it to prevent security breaches. Moreover, the lesson learned from the Twitter and Dropbox cases is that the vulnerabilities may not be inherent to the 2FA concept itself, but to the features the applications implement. Therefore, MFA does not add more security if the whole application workflow is not reviewed and adapted accordingly. Despite having a few disadvantages, such as the requirement of having a physical token around for the *something you have* factor, the risks of not adopting MFA may prove to be more costly.

Chapter 4

Secure User Authentication in Cloud Environments

This chapter enumerates various good practices for building stronger authentication procedures with basis on the issues discussed in the previous chapter of this dissertation. A model for making authentication more robust is also proposed, along with a prototype showing its applicability to cloud computing.

4.1 Introduction

The previous chapter described several security issues impacting the cloud computing model and authentication. The security state of cloud computing is still puzzling. It shifts computing perceptions entirely from the customer viewpoint, outsourcing IT systems instead of adopting local solutions. Authentication is also being subject to changes. The static one-factor login is not longer viable. Systems are often vulnerable to data breaches, and attackers are increasing their arsenal of tools and techniques for brute-forcing password hashes, gradually lowering the reliability of password usage. As such, research on authentication is focused on augmenting existing mechanisms or utilizing QR encoding, SMS messages, OTPs, or biometrics based on growing mobile technology for devising novel MFA approaches, while aiming to achieve important security properties like perfect forward secrecy and completeness. Nonetheless, little attention is payed to the underlying infrastructure, and few have looked into harnessing the cloud computing technology for the purpose of making authentication more robust, in response to the adversity of Internet threats to public clouds, particularly to the management interfaces. Section 4.2 overviews and describes a general model for implementing strong authentication procedures inspired on the cloud computing related issues and technologies. Subsequently, Section 4.3 overviews and describes the prototype built upon that model using public-key cryptography. Finally, Section 4.4 hands out general recommendations for achieving secure user authentication. The following discussions are partially based on the contents of the third [SFG+14] scientific publication of the research work performed throughout this masters program.

4.2 The Proposed Model

Management interfaces of cloud computing services are exposed to a subset of issues that affect the Internet. They are also typically accessed by several customers and can be scrutinized by attackers outside the cloud network for public clouds. As such, the model presented in this

section aims to address the requirement of making authentication more robust at the frontier of the cloud network.

4.2.1 Goals of the Model and Assumptions

Generally speaking, the proposed model aims at minimizing the impact of the Internet threats pointed out in this dissertation while focusing on reducing the openness of cloud environments. For this, the proposed model builds upon the worst case scenario, comprised by public cloud environments. Other types of cloud will be more secure and thus the model applies to them also. Components forming a public cloud include standard web technology and communications over the Internet using HTTP or HTTPS. Given that expecting everyone to adhere to a strict means of building a procedure is not realistic, the model should be characterized by an open approach in the sense that several authentication schemes can be deployed over the devised solution, functioning like an underlying incubator. Essentially, the model should represent a skeleton that flexibly incorporate one or more authentication mechanisms based, e.g., on public-key cryptography, ZKPs, MFA, and others, but also while considering SSO and passwordless trends and without setting aside password-based authentication, so as to assure backward compatibility.

Because cloud computing brings benefits like elasticity and resource pooling, the model should take that into account also. In this case, one of the design objectives of the model is to take advantage of that elasticity to benefit both the provider and the client, since the technology can be harnessed to build securer and more robust infrastructures, particularly for authentication. This is the case of the proposed model. By being based on IaaS cloud computing, the model becomes flexible enough to incorporate the desired features by means of pre-packaged template images or VM disks. On the side, cloud instances can be booted up with little overhead, and thus availability should be upheld high. In addition, the model complexity should be low since virtualization allows configuring VMs on-the-fly.

It should be noted that the proposed model does not portray a new approach for authentication nor consists of a new scheme combining the several technologies. Instead, it aims at providing an underlying framework for building authentication methods, while lowering the risks of public cloud infrastructures by narrowing down and restricting the attack vectors scope. It hence assumes that a secure cloud computing environment with regard to IaaS and virtualization is already in place, despite the issues discussed in subsection 3.3.1.

4.2.2 Overview

Part of the model proposed in this dissertation for secure user authentication is inspired in the architecture of the Whonix [Who] OS. The advent of virtualization, and particularly of IaaS clouds, brought new possibilities for dedicated computing. This is the case of Whonix, a Debian-based OS that consists of two VMs connected one to the other in a private virtual

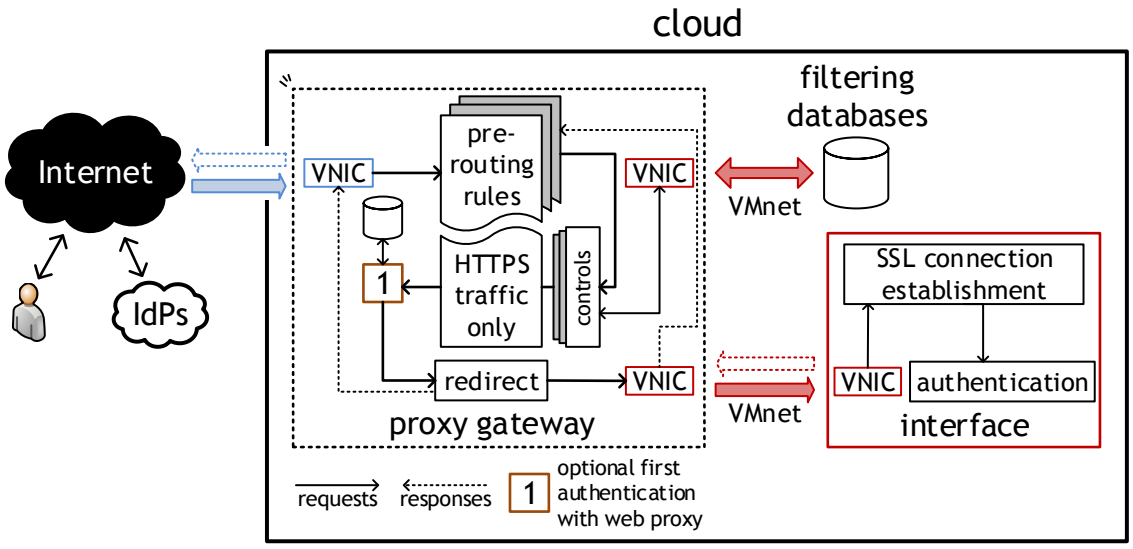

Figure 4.1: Proposed model for secure user authentication on cloud management interfaces.

network, each one having its own purpose. One of the VM acts as a *gateway* for the other, which is called the *workstation*. All network traffic and communications originating on the workstations is first forwarded to the gateway that, in turn, routes packets through The Onion Routing (Tor)[1]. Because the gateway acts as both entry and exit points, it should be the first one to become compromised in the hypothetical case of an attack. By restricting the scope of the VM exposure to the outside, the attack vectors are therefore limited.

The schematics of the proposed model are depicted in Figure 4.1. An implementation of the model is suitable to place at the perimeter of cloud networks for mediating access to management interfaces and other services, particularly for user authentication. As in the Whonix architecture, the model stresses out the importance of a dedicated gateway to proxy traffic through, but with additional capabilities. Such capabilities fall within the cybersecurity perspective, regarding standard security controls useful for fighting everyday threats. From here on, the gateway is called *proxy gateway*. The proxy gateway runs on a VM, protecting the inner interface that is hosted by another VM. Preferably, the proxy gateway acts like a transparent proxy, hiding its presence to the outside. Both VMs are connected in an isolated and secure manner.

When a packet arrives at the proxy gateway, some pre-routing tables can be applied before passing on the packet to kernel routing decisions. This is useful for changing the destination IP address in order to forward traffic flows to a desired network node. Nonetheless, the security controls step in before forwarding any packet. The security controls can include firewalls, IPSes or IDSes. For HTTP traffic, Web Application Firewalls (WAFs) are particularly useful for inspecting HTTP requests going to a specific web application, preventing SQLi or XSS attacks

[1]Tor is an anonymous P2P network for Internet communications. Its goal is to hide the real IP address of a certain Tor client by routing packets within Tor in an unpredictable and secure way through distinct paths. It can be used to bypass firewall filters in oppressed countries or to evade surveillance, for example.

with basis on set of policy signatures. In such case, it would be necessary to deploy a particular web proxy in the proxy gateway since, for maximum security, SSL/TLS should be enabled in HTTP, so as to assure communications confidentiality and integrity. In addition, an optional first factor of authentication may be setup within the proxy gateway. For example, a general certificate could be used to authenticate therein, and then specific user authentication by means of a second factor username and password would identify the user on the interface, for which different privileges can be applied in the application logic. Secure, isolated private virtual networks connect both the VMs by configuring VNICs appropriately. More VNICs can be added easily to connect the proxy gateway to filtering databases useful for security purposes. Users can authenticate via standard authentication methods implemented by the cloud provider of via IdPs.

There is also the possibility of creating personal authentication infrastructures, customized according to the requirements of very particular customers. For customers adopting various SaaS or PaaS applications and IaaS systems, it may be useful and suitable to allocate a set of resources for more proxy gateways. On the enterprise side, firewalls can be configured to restrict egress and ingress traffic points to solely communicate with those proxy gateways. Essentially, the cloud infrastructure would act as a big proxy for the customers with a ranged set of security tools. This line of though coincides with the work of Salah *et al.* [SACZ+13] and with what McAfee mentions in [Str13].

4.2.3 An Overlay Cloud Network

Salah *et al.* [SACZ+13] provided a study proposing an horizontal S_{ec}aaS solution spanning the majority of the areas with regard to security. The proposal consists on a security solution for enterprises that sets up an overlay cloud network functioning like a big proxy from the perspective of the customers networks. The overlay cloud network would take advantage of IaaS cloud computing to deliver on-demand IaaS security services that would comprise standard security controls, namely Distributed Denial of Service (DDoS) prevention and protection, IDSes, firewalls or IPSes, anti-spam, anti-phishing, and malware analysis and detection. A frontend security center within the overlay network would provide management capabilities for incorporating company policies, and features for integrating with the SIEM process. Moreover, the overlay network can implement additional servers for proxying specialized traffic. Such additional servers can depict SSO services for augmenting company authentication with internal AD or LDAP servers. A scalable load balancer is put on the network input point, while output traffic is forwarded to customers, thereby making the overlay network acting like a big proxy.

The overlay cloud network adheres to the hybrid cloud deployment model. In their case, enterprise communications were required to be routed through that point before being forwarded to the destination, therein being subject to all the security controls within the security cloud.

On the customers end, incoming traffic is firewalled and only traffic coming from the security cloud would be allowed. Their tests depict advantages that include network concealment against reconnaissance techniques, detection and prevention effectiveness, flexibility for additional resources, higher performance, and costs reduction. Part of the model proposed herein elaborates on the same advantages of an overlay cloud network, though at a smaller scale and for the purpose of authentication. Additionally, it is devised to be placed at the edges of cloud networks only, close the management interfaces.

4.2.4 Analysis of the Proposed Model

The proposed model enjoys various advantages from both the security and efficiency viewpoints. The most noticeable one is that the interfaces that are to be accessed by clients are pushed further back into the network, concealing them better from the outside. Internal network information of security appliances and production networks are better protected against reconnaissance attacks by using the architecture of the proxy gateway. Since the proxy gateway can be under high loads of network traffic, it can be more susceptible to attacks targeting the available bandwidth or the processing power to achieve DoS states, using flooding to create traffic jams or processing constraints. In such cases, hypervisors can try to allocate more resources. If the VM crashes, then it is easy to boot up another of the same image template with little overhead and within a few seconds or minutes. The same VM disk can be used for new VMs, thereby avoiding losing data. Since virtualization allows to easily make backup snapshots of VMs, there is also the possibility to keep updated copies for these situations. The security controls depicted in the model may be embedded within the proxy gateway through Security Virtual Appliances (SVAs). A SVA is a pre-packaged security software that simulates a blade server[2] of some sort in a virtualized way. When processing packets, the filters can be sourced from a diverse set of databases regarding cybersecurity. For example, IP addresses, URLs or domains with a bad reputation (*e.g.*, blacklisted in spam lists or belonging to malware CnC servers) can be ruled out on-the-fly. Such an approach would mitigate or at least decrease the odds of having well succeeded attacks using botnets to brute force vulnerable interfaces like the Plesk Panel. The management interfaces themselves should be built by following good practices in web development, while avoiding ads and other channels of malware spread. Furthermore, by using point-to-point private virtual networks, the isolation of network traffic is ensured. On top of all this, HTTPS is mandatory for securing communications. But one should pay attention to the mixed content risk [Tro13]. Some websites deploy HTTPS in a wrong manner when clients fetch for other resources like images over HTTP within the same session. This vulnerability should be avoided.

For multiple gateway proxies spread across the network frontend, a proper load balancer would

[2] A blade server is a server computer with a modular design that is stripped down from the main body of an appliance chassis. A blade server is usually specialized in a given functionality to increase the number of features of the appliance.

help dilute a packet bombardment to ensure adequate traffic distribution among the proxies, thereby diminishing the effectiveness of DoS attacks. In addition, these measures may also comply with company policy regarding disaster recovery plans. Virtualization abstracts VMs from the underlying hardware on which they run. This is good for compatibility purposes. Therefore, it is possible to package pre-defined image templates that deploy specific features. A diverse set of those image templates can be built to offer flexible service plans to customers, varying prices accordingly with security levels. With this, particular authentication infrastructures can be setup for customers who want flexible strategies.

The possibility of firing up VMs for proxy gateways on demand provides also for the possibility of using fault-tolerance techniques (e.g., byzantine fault-tolerance [VCB+13]) to deal with the possibility of having one or more VMs under the control of an attacker. The proxy gateways may be set up as hardened OSes by stripping out unnecessary services and kernel modules and by installing only security software.

4.3 Prototype for the Proposed Model

The prototype described in this section serves as a PoC for the model proposed in the previous section. The main objective is thus to provide an example of how the model can be applied to make authentication more robust on public environments while resorting to public-key cryptography, smartcards and readily available and open source technology.

4.3.1 Overview

An illustration of the prototype is shown in Figure 4.2. The technologies used to build it may be summarized as follows:

- A local experimental IaaS cloud was engineered by means of VMware products for virtualization, namely VMware Fusion and Workstation, both well-know hosted hypervisors for Mac OS and Windows OSes, respectively. The experimental cloud was composed of two VMs, one for the proxy gateway and the other for the management interface.
- The proxy gateway was configured with `iptables`, the Linux kernel firewall, so as to drop and redirect the respective traffic to the management interface transparently. It ran the 64-bit version of Ubuntu 13.04.
- The management interface was simulated with the Apache web server particularly for HTTPS support. A self-signed certificate was issued to the server by means of OpenSSL. It ran the 64-bit version of Ubuntu 13.04.
- The Portuguese identity card was utilized to carry out mutual and strong authentication to the management interface. The smartcard carries an authentication certificate protected by a PIN. This technology was used to complete SSL handshakes for HTTPS by exchanging

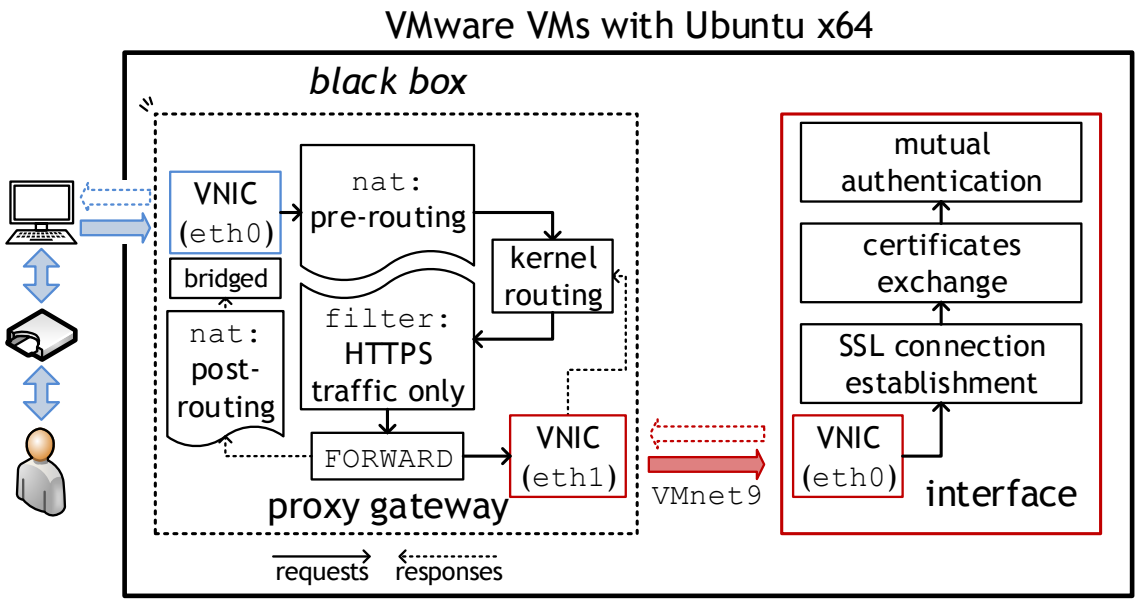

Figure 4.2: Prototype based on the proposed model for strong authentication using the Portuguese identity card for mutual authentication.

client and server certificates. A common smartcard reader was used for communicating with the Portuguese smartcard.

- Finally, the Mozilla Firefox web browser was used to configure the middleware of the Portuguese identity card, so as to be used as a fat client to access the management interface.

The Portuguese identity card is a smartcard issued by the Portuguese government to each citizen, containing from factory two certificates useful for public-key cryptographic applications. The certificates unequivocally identify a person and, therefore, is perfect for authentication purposes. By utilizing a certificate on the interface side, the prototype enjoys strong and mutual authentication, particularly in the SSL handshake. Mozilla Firefox was a suitable browser for the task at hands, since it was needed to install the middleware of the Portuguese identity card (more on this in subsection 4.3.3). The experimental IaaS cloud isolates communications between the VMs by means of the private virtual network `VMnet9`, in this case. Although both VMs ran Ubuntu for exemplification purposes, a real deployment may immediately use an existing hardened Linux appliance distribution like IPFire [Lig], which may be served over a virtual environment. Since `iptables` is capable of packet analysis and redirection, it seemed suitable to use in the proxy gateway by means of packet filtering rulesets. The `iptables` utility applies rules from particular tables in a predetermined order, which affect how each packet is processed by the kernel. In the prototype, the most useful tables were `nat` and `filter`, each one containing rulesets in chains that are applied as the packets traverse the VM. Each of these pieces are better described in subsection 4.3.3, which explains how each one fits in the puzzle.

4.3.2 The Portuguese Identity Card

The Portuguese identity card [Pord] is a cryptographic smartcard capable of performing cryptography operations in a simple, secure and almost transparent manner to the user. The Portuguese government issues these cards ad-hoc. They contain personal information about the citizen on the surface of the card itself and in digital format as well, stored in a memory unit. Assuming that the government is a trusted third-party (NSA revelations apart), the data withhold by the card unquestionably is truthful and identifies a person. Like any other identity card, they come stamped with a validity date of a few years. The most interesting fact about the smartcards is that they embed the cryptoprocessor and two 1024-bit RSA key pairs and associated certificates each, which are intended for use by its rightful owner. One of the certificates is destined for signing purposes, while the other is intended for authentication. The private keys are dully protected on flash memory by two PINs, one for each certificate. Both certificates expire precisely in the day the identity cards become invalid as well. The validation chain of the certificates includes signed certificates belonging to the Portuguese government, available from the oficial site, and certificates issued by a world class certification authority. The system relies thus on a PKI [Porb] to provide trust and define their usage. This PKI supports the validation of signatures or authentication, manages valid or invalid cards (*e.g.*, using revocation lists), and specifies the issuance of new ones or re-issuance of expired cards and certificates. The Online Certificate Status Protocol (OCSP) is utilized for checking the revocation status of certificates on the PKI. A middleware with a simple interface is available [Pord] for reading some of the information stored on the card. Access to sensitive data requires the insertion of the respective PIN. The cryptographic utilities of the Portuguese identity card can be accessed through standard APIs for cryptographic smartcards in C or Java [Porc]. Nonetheless, for the purpose of building this prototype, such APIs are not needed.

4.3.3 Prototype Configuration

The configurations discussed in this section detail how the prototype testbed was mounted. Each configuration and setting is discussed thoroughly to better justify why and how each component fits in the prototype. These allow to better perceive how authentication can be made securer right from the start.

4.3.3.1 Cloud Testbed

The local cloud setup was composed by two VMs running co-resident on the same physical machine. Both were instantiated with the default configurations regarding the VCPUs, memory size, among others, except for the VNICs. Typically, hypervisors exhibit a functionality that allow one to define a new type of virtual network, named `VMnet` followed by an identifier number in VMware products. The three basic virtual networking types are already defined by `VMnet0`,

VMnet1, and VMnet8, referring to bridge networking, host-only and Network Address Translation (NAT), respectively. The bridge mode allows to wire up VMs directly to the real, physical Ethernet network. Host-only creates an isolated private network with the host, making Internet unavailable to the VMs. Finally, NAT shares the IP address of the host to allow connectivity to the outside. Hypervisors create virtual network adapters on the host, virtual Dynamic Host Configuration Protocol (DHCP) servers and other devices for these networking purposes.

For building this prototype, a custom network named VMnet9 was created. The type of the network is host-only and the local DHCP service was used to distribute IP addresses of the address pool 10.0.0.0/8. The proxy gateway VM was configured to use two VNICs, while the interface VM used only one. The eth0 VNIC of the proxy gateway was networked through a bridge. It represented the entry point of traffic coming from the outside, while the eth1 VNIC connected to VMnet9. On the interface VM, the eth0 VNIC was connected to VMnet9.

4.3.3.2 Mozilla Firefox Device Manager and Portuguese Identity Card Middleware

The Mozilla Firefox is one of the most popular web browsers nowadays, along with Internet Explorer and Google Chrome. For testing purposes, Firefox was found to be the most suitable browser, mostly because it was needed to add the middleware of the Portuguese identity card into the browser. From the three aforementioned browsers, Firefox was the one on which the installation worked most of the times. Firefox is robust and rich in functionality and features. In this case, the feature of interest is the device manager, accessible via the Advanced tab of the options dialog, and therein choosing the Certificates tab and finally clicking on Security Devices. After the last click mentioned, another dialog window opens up, containing the various security modules and devices integrated with the browser.

The device manager allows to add new security modules and devices easily. In the Portuguese identity card case, it is first needed to install the middleware for the underlying platform[3]. The installation of the software follows a standard workflow without requiring configurations for Windows and Mac OS platforms. Once installed, the middleware is ready to be configured on the browser. On Firefox, this is achievable by loading a new module with the pteidpkcs11.dll DLL file belonging to the middleware. Every time the smartcard reader detects the Portuguese identity card, the certificates are immediately loaded into the device manager, allowing to use them for any web specific purpose. The security modules must follow the PKCS#11 [RSAb] standard. The PKCS#11 standard specifies a device-independent API to the cryptographic capabilities of any type of cryptographic device, such as a smartcard. The series of Public-Key Cryptography Standards (PKCSes) are published by the RSA Laboratories.

[3]The Portuguese identity card is available for popular platform flavors [Pord], namely Windows, Mac OS and Linux Ubuntu.

4.3.3.3 Proxy Gateway

The proxy gateway is configured by means of `netfilter`, the default packet filtering framework shipped with the Linux kernel. The user-space applications for `netfilter` are the `iptables`, `ip6tables`, `ebtables`, and `arptables` utilities, used for handling Internet Protocol version 4 (IPv4) packets, Internet Protocol version 6 (IPv6) packets, Ethernet frames, and Address Resolution Protocol (ARP) packets, respectively. Perhaps the most important features of `netfilter` are the stateless and stateful packet filtering, the network and port translation, and the connection tracking mechanism. Within the scope of the subject at hands, the one of interest is the stateless packet filtering for redirection purposes. Since the goal of the prototype is to show authentication on online services over the Internet, the suitable utility for writing packet filtering rules is `iptables` (assuming IPv6 is not implemented).

The core of `netfilter` is constituted by several tables that are available in full or in part to `iptables`, `ip6tables`, or `ebtables`. Each table is broken down to various chains that ultimately contain the rules. Each chain is only swept for rules on specific parts of a packet and on specific timings of the traffic flow (*e.g.*, when the packet is entering -INPUT- or exiting the kernel -OUTPUT). The main tables of `iptables` are `filter` and `nat`, each one depicting a few chains. The `filter` table contains the `INPUT`, `FORWARD`, and `OUTPUT` rule chains, while the `nat` table discriminates the `PREROUTING`, `OUTPUT`, and `POSTROUTING` chains. From `filter` table, all chains were used when constructing the prototype, while `PREROUTING` and the `POSTROUTING` chains where the ones used for the `nat` table. `PREROUTING` rules are applied before the packets are submitted to any input routing decision, while `INPUT` or `FORWARD` rules apply after such a decision. Forwarded packets are not delivered to any local process in the application layer, which means that packets never pass over the network layer. In the same process, `OUTPUT` rules apply after a kernel routing decision, and then `POSTROUTING` comes in. By default, `iptables` adds rules to the `filter` table, if no other table is specified otherwise with the `-t` flag (*e.g.*, `-t nat`).

The proxy gateway is configured as a *black box*, denying all network traffic except HTTPS traffic destined for port 443. This effect is achieved by setting the global policy (`-P`) as `DROP` in the `filter` table with the following command:

```
$ iptables -P INPUT DROP; iptables -P OUTPUT DROP; iptables -P FORWARD DROP
```

.

At this point, all traffic is disallowed by the firewall. To allow HTTPS, port 443 must be opened in both `INPUT` and `OUTPUT` chains for the incoming requests and outgoing responses. The following commands do that, respectively:

```
$ iptables -A INPUT  -p tcp -s 0/0 --dport 443 -j ACCEPT
```

,

```
$ iptables -A OUTPUT -p tcp -s 0/0 --sport 443 -j ACCEPT
```

,

where `-p` specifies the transport protocol, `-s` the source address, `--dport` and `--sport` the destination and source ports, respectively, and `-j` the rule action. The next step is to forward the traffic to the VNIC belonging to the interface VM, and the reciprocal as well. Before that, the Linux kernel must be told to allow forwarding IPv4 traffic. This is achievable with the following commands:

```
$ echo 1 > /proc/sys/net/ipv4/ip_forward
```,

```
$ sysctl net.ipv4.ip_forward=1
```.

The VNICs interfaces (*i.e.*, `eth0` and `eth1`), shown in each VM on Figure 4.2, are the ones considered in the following rules. For testing purposes, the rules were loosen up to allow traffic from the broad 10.0.0.0/8 private subnet setup between the two VMs. This does not impact security since the rules also include the input (`-i`) and output (`-o`) Ethernet interfaces of the `FORWARD` rules, and thus they are specific enough because the VNICs are securely connected through the virtual network for those Ethernet interfaces. The `ACCEPT` rules in the `FORWARD` chain in `filter` are the following:

```
$ iptables -A FORWARD -i eth0 -o eth1 -p tcp --dport 443 -s 0.0.0.0/0  -j ACCEPT
```,

```
$ iptables -A FORWARD -i eth1 -o eth0 -p tcp --sport 443 -s 10.0.0.0/8 -j ACCEPT
```.

What is left to be done is to transparently proxy traffic through. HTTPS requests are actually made to the proxy gateway which needs to forward them on to the hidden VM. This can be done by changing the destination IP address of the requests before letting the kernel deciding about its routing, and by changing back the source IP address of the responses to the bridged VNIC of the proxy gateway. For the IP addresses 10.128.0.1 and 192.168.1.2 belonging to the interface VNIC and the bridged VNIC of the proxy gateway, respectively, the following rules in `PREROUTING` and `POSTROUTING` would produce the desired effect:

```
$ iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 443 -j DNAT
--to-destination 10.128.0.1:443
```,

```
$ iptables -t nat -A POSTROUTING -o eth0 -p tcp --sport 443 -s 10.0.0.0/8 -j
SNAT --to-source 192.168.1.2:443
```.

4.3.3.4 Management Interface

The work on the server side includes configuring the web server and the SSL for mutual authentication purposes by means of public-key certificates. For demonstration reasons, the OpenSSL cryptographic suite was used to generate a self-signed certificate that is assigned to the server. First, the private key of the server as well as a certificate request can be generated with the following commands, respectively:

```
$ sudo openssl genrsa -out private.key 2048
```,

```
$ sudo openssl req -new -key private.key -out request.csr
```,

where `private.key` is a 2048-bit private key and `request.csr` is the certificate request. Then, the simulation of the generation of a signed certificate in a CA can be achieved with the following command:

```
$ sudo openssl x509 -req -days 365 -in request.csr -signkey private.key -out
certificate.crt
```

, from which the certificate `certificate.crt` is obtained with a validity of 365 days, in this case. The web server used was Apache, more specifically the `apache2` daemon running version 2.2.22. Apache is installed via standard repositories. In Ubuntu, it may be installed by issuing the following command on the terminal:

```
$ sudo apt-get install apache2
```

.

Apache configuration files are in `/etc/apache2/`, while the default path for the web page is in `/var/www/`. To activate the SSL module (`ssl_mod`) on Apache and turn on HTTPS support, the following commands are required to be inserted on the command line:

```
$ sudo a2enmod ssl
```

,

```
$ sudo a2ensite default-ssl
```

.

After the previous step, one needs to configure SSL accordingly. The configuration file is `/etc/apache2/sites-available/` and is named `default-ssl`. Mutual authentication is activated by requiring to verify the client by means of a certificate. The verification depth of the certificate path validation chain is set to a high number so as to go through the entire validation process, for which the intermediate and top certificates are in `/etc/ssl/certs`. The certificates therein stored are shared with Firefox on Linux systems. By assuming that the server private key and corresponding certificate are in `/etc/apache2/`, the following configuration directives of `/etc/apache2/sites-available/default-ssl` achieve these aforementioned purposes:

```
SSLVerifyClient require,
SSLVerifyDepth 999,
SSLCertificateFile /etc/apache2/server.crt,
SSLCertificateKeyFile /etc/apache2/server.key,
SSLCACertificatePath /etc/ssl/certs/.
```

In addition to the above directives, it is also possible to pass SSL items and environment variables to various contexts, namely the Hypertext Preprocessor (PHP). Amongst such items are the Privacy Enhanced Mail (PEM)-encoded certificates belonging to the connecting clients. This detail is useful to the web application logic, so as to uniquely identity a user registered beforehand. PHP 5 can be installed with the next command:

```
$ sudo apt-get install php5
```

.

PHP was used herein to build the logic that emulates the authentication of a client to a web application. The idea is to ask a user to first authenticate using the citizen card (proof-of-pos-

session) by means of the respective certificate and insertion of a PIN (proof-of-knowledge), and then provide access to the application if the name of the user in the certificate that arrived to the server concerns a registered used of the application. In this prototype, a PHP snippet was used to get the name of the client from the certificate, which can be used to compare it with a database of registered users, for instance. The registration would be conducted by simply carrying out normal authentication. The system would recognize the first time a user logins by simply comparing the received unique identifier with the all identifiers of the current database. If a match would not be found, then the user could be prompted for confirming the registration. Notice that this system does not use usernames or passwords, thought the authentication is strong and uses several factors (possession and knowledge). In the web application file `/var/www/index.php`, PHP has access to such information in the `$_SERVER` global variable. Specifically, the `$_SERVER['SSL_*']`, where the asterisk is replaced by some string denoting one SSL variable passed to PHP. For example `$_SERVER['SSL_CLIENT_VERIFY']` tells whether or not if the client sent a certificate.

Finally, for the prototype to work, it is needed to tell Apache about the intermediate and top certificates in the certificate path validation chain of the authentication certificates included in the Portuguese identity card. A total of four certificates compose the certificate path validation chain. Those certificates are available at [Porb]. However, they are in CER format. Apache requires them to be PEM encoded, to which it is possible to convert with OpenSSL by means of the following command:

```
$ openssl x509 -in certificate.cer -inform DER -out certificate.pem -outform PEM
```

.

After moving the resulting PEM-encoded certificates to `/etc/ssl/certs/`, one needs to rehash the path so as to enable Apache to recognize and trust the new certificates. OpenSSL provides an automatized script named `c_rehash` for this, which consists of the following:

```
$ sudo c_rehash /etc/ssl/certs/
```

.

The final step consists in defining the default gateway address of the management interface by pointing it to the proxy gateway `eth1` VNIC. For an IP address of 10.128.0.0 assigned to the `eth1` VNIC, the following command adds a default route to the Linux kernel routing table:

```
$ sudo route add default gateway 10.128.0.0 metric 0 eth0
```

.

4.3.4 Analysis of the Prototype

The functioning of the prototype is simple. The user inserts the Portuguese identity card into a smartcard reader that is connected to a computer with the Firefox browser configured with the respective middleware. Then, the target web interface is accessed for the first time with HTTPS on. Since HTTPS is turned on, the browser will ask for a certificate on the user side after verifying the certificate belonging to the server, exchanged at the beginning of the connection. Firefox asks the user to choose the certificate. At this point, the certificate is selected and a

few moments later the corresponding PIN is asked to be inserted by the smartcard middleware. If Apache is able to validate the received user certificate and if the web interface successfully performs some operation at the application logic level with basis on the data contained in the certificate, the user is successfully authenticated in an 2FA approach. The authentication certificate contains unique information in the subject field identifying the associated citizen. Such information can be used to register a user and identify it in subsequent logins. For example, the subject field is populated with the full name of the citizen and a unique number assigned by the Portuguese government to identify each citizen. The user proves to have the smartcard around and further proves to know to associated PIN. This mechanism combines *something you have* with *something you know*, similarly to banking cards.

The mechanism implemented in the prototype can be further extended by leveraging the convenience of having two certificates for distinct purposes. The management interface can, at any time, send a random nonce for signing purposes. In such cases, the smartcard would be needed to be inserted at those moments and the user would have to know the correspondent PIN, so as to access the private key for creating a digital signature. The server receives and verifies the signature of the nonce and validates the response. Such approaches would not degrade much the user experience and they would increase security, since the signing PIN may be different from the authentication PIN. For more critical operations within the management interface, like creating or stopping VMs, the server can just keep on asking to sign nonces. Even if the smartcard is lost or forgotten within the smartcard reader, an adversary would still need to know the PIN to successfully access the system. Brute-force would hardly be an option since the smartcard gives a limited number of tries to guess the correct PIN, blocking the card afterwards. This also applies for changing any of the PINs.

There is no need for restricting traffic on the server side. The proxy gateway VM already filters out everything else. This puts more processing overhead on the proxy gateway side. In the hypothetical case of a DoS scenario, the server is relieved from the constraint put by the `iptables` rules. But, in any case, cloud computing technology can just allocate more computing resources on-the-fly to the proxy gateway. If the VM goes down, another one can be booted up with little overhead. Other instances of the prototype can be setup. For example, a web proxy installed in the proxy gateway can serve as a middleman for the authentication interface, performing HTTP acceleration or reverse proxy, and perhaps a first factor of authentication. If successful, the user is then redirected for the interface and a new HTTPS session is established. Such would benefit backward compatibility and scalability, since the legacy authentication based on passwords may still be used with little configuration burden on existing systems.

4.4 Recommendations for Secure and Transparent User Authentication

This subsection enumerates a few recommendations or good practices regarding user authentication on cloud environments. These recommendations elaborate on the technologies used, but also around the protocols, and how simple measures can be adopted to make the procedures more secure.

Although QR codes can be maliciously crafted with shortened URLs hiding a compromised location on the Internet, they represent an useful resource. QR codes are easily and efficiently encoded and decoded and their usage is increasing, motivated by the rise of mobile technology. For now, MFA is greatly based on 2FA by means of OTPs that are required to be inserted after username and password verification. This second factor or the entire mechanism could be easily replaced by a procedure that uses QR codes, which are easier to use, making the process more seamless and enjoyable and eliminating the burden associated with keyboard based inputs that sometimes have to be repeated several times a day. Risk-based authentication would also come in handy for those cases. Public-key cryptography schemes can be mixed with QR codes for exchanging cryptographic data. This is the case of Authentify xFA [Aut] (presented in sub-subsection 2.5.4.3), which encrypts QR codes to exchange biometric data and utilizes a PKI as a backbone for security based on certificates. One of the good things about this approach is that the browser on the computer automatically logins without further interaction with the user, beyond authenticating via voice recognition. Google tried it first and then came Authentify. One of the problems is that biometric signatures are stored on servers belonging to Authentify. The other issue is that the mobile device keeps key pairs belonging to the user.

Turning a smartphone into a personal authenticator seems to be a convenient and almost certain trend, since these devices are increasingly more powerful and capable of storing unique authentication data like passwords or keys. But using the single device for authentication comprises only a single factor *proof-of-possession* [MFH11] procedure. As such, a PIN should complementarily be used to protect that kind of data, just like what happens with smartcards. Essentially, the PIN would serve to decrypt the data, and thus malware and some attacker who gained physical access to the device could not authenticate on the behalf of the user. The smartphone approach coincides with the friendly user experience that most employees nowadays want. The ease in connectivity urged users to carry at all times their own devices for social networking and other personal matters which caused the BYOD phenomenon to settle in. But, for enterprises, the BYOD approach brings security challenges, since the company policies may not be enforced on the devices belonging to the employees. Enterprises who give priority to security may want to check upon tokens or change passwords every 30 to 90 days. Finally, more rigorous authentication processes should be deployed on systems where data is more sensitive or confidential.

This is the case of biometrics, but in physical access environments.

Cookies and HTTP form the basis for web-based communications and sessions with web servers. As discussed in subsection 2.3.4, the Channel ID protocol cryptographically bounds cookies with the underlying TLS channel. This approach seems suitable for avoiding cookie theft, and therefore eliminating MitM threats. It is also recommended to follow good security practices while browsing the Internet. In the first place, suspicious websites should be avoided to prevent malware infection. Secondly, security should be enforced. HTTPS can be enforced by installing addons on the browser. HTTPS Everywhere [Ele] and Force-HTTPS [JB08] are good examples of such addons. Thirdly, the user should play a more distinct role in security. For example, at Amazon EC2, VMs are accessed through an SSH tunnel. The user is authenticated by the servers by proving to have the correct private key for a particular VM. The corresponding key pair can be generated offline by the user or at the EC2 cloud side. OpenStack offers the same possibility (to generate the keys on the server side), unlike, e.g., the MEGA cloud service, which collects entropy on the user end for generating strong cryptographic keys directly on the client side. Apart from possible processing constraints, which do not apply in most cases nowadays, it seems more secure that a key pair for authenticating a user should be generated at the user side, since the private key does not need to be known by the server in any moment of the registration or authentication procedures.

The Google Authenticator or other similar software to implement on the Linux PAM stack also seems like a good approach for stronger authentication, despite its drawbacks [Rya12]. The PAM stack can be configured to ask for OTPs when users issue commands with root privileges and when authenticating to the OS. For IaaS clouds, this second layer of authentication is definitely a securer approach for remote access to VMs. The recommendations for systems using password-based authentication rest on the premises of prevention. That is, frontend websites should be developed with emphasis on security by following a rigorous Software Development Life Cycle (SDLC) [Mar13] to avoid vulnerabilities. Moreover, passwords should never be stored in plaintext on backend databases that connect to the main frontend applications. Instead, they should be salted and hashed down. It is common practice to adopt popular cryptographic algorithms like MD5 and SHA-1 for hashing purposes. But the problem with these algorithms is that they were designed to be plain fast while providing the security properties wanted for cryptographic hash functions. This practice eases the task of crackers. When a data breach of hashed passwords takes place, attackers can brute-force through them all more rapidly since the same hash functions are faster to execute. Instead, it should be used cryptographic hash functions that provide robust security properties as well, but that on the side are extremely slow (in the cryptographic sense). Just a few seconds of delay exponentially increases the cracking time for attackers. This is the case of `bcrypt` [Hal10], that modifies the Blowfish keying schedule to make a slow hashing function.

One last recommendation must necessarily point to the fact that there is an incredible wide panoply of authentication methods described in the literature, some of which enjoy a mathematical proof and low overhead in terms of user intervention. These include public-key and ZKPs. The ones that may be less user friendly can now be combined with modern technologies, *e.g.*, smartphones and QR codes, for seamless user interaction, which removes that constraint out of their disadvantages. Additionally, the computational complexity of such methods is less of a problem nowadays, since processing units and memory components increasingly got faster and cheaper. Nonetheless, some of these methods are still neglected in favor of insecure authentication procedures, perhaps motivated by lack of knowledge of developers in the field or because these procedures are simpler to implement, which should not be a reason in this case. The recommendation is that team leaders should start motivating developers into searching and using better mechanisms, looking for ways of integrating and making the authentication simpler to the user and, at the same time, stronger, thereby adhering to rigorous SDLCs.

4.5 Conclusions

This chapter has described the proposed model for secure authentication on cloud management interfaces, coupled with a PoC prototype showing its applicability. Throughout the text it was emphasized how security should be employed in order to build a secure authentication procedure in general. The last section pointed out several several good practices for achieving secure and more transparent user authentication. Authentication is generally an undermined component of IT systems, especially on online services that develop deficient applications with loopholes and backdoors, allowing attackers to access restricted areas. The degradation of the well-established static login no longer fulfills the security requisites of the everyday life of an Internet user. As workarounds, the academia and the industry are focusing on new authentication methods. In general, MFA brings advantages from a security viewpoint, but not in a perfect way. Such new approaches require implementation to be executed thoroughly to avoid flawed authentication, and also more robust models and infrastructures as showed throughout this dissertation. The model proposed herein aims at minimizing the dangers coming from the outside world and was specially designed for cloud environments, since it depends on virtualization techniques for some of its characteristics. One of the main advantages of the model is that it backward compatible, since enables one to deploy new authentication mechanisms over existing ones.

The presented prototype instantiates the aforementioned model using only readily available and open source technologies, to show the feasibility of the implementation. Actually, one of the problems pointed out for authentication in computer systems is that there are readily available secure solutions described in the literature which are nonetheless neglected in favor of less secure or even obsolete mechanisms, for reasons of complexity, costs, or overhead. The presented prototype uses strong and mutual authentication by means of public-key cryptogra-

phy, resorting to cryptographic material that, for example, Portuguese citizens already have in the Portuguese citizen card. The main objective of this prototype was to build a PoC, but it surely could be improved and thoroughly tested, which is left as a future line of research.

Chapter 5

Conclusions and Future Work

This chapter includes a few comments on the evolution that the perception regarding authentication requirements and mechanisms is currently undertaking, and how that will revolutionize authentication in the future. Afterwards, the main conclusions of the work presented in this dissertation are drawn, and then possible directions for future work are handed out.

5.1 On the Horizon

Authentication has been largely based on the static password approach for as along as computers existed. For that many years an association between an identity and a person has been achieved by combining usernames and passwords. It was up to each person to derive some strong password that would resist brute-forcing algorithms. But the evolution of technology, as foreseen by Moore's Law[1] in terms of processing power has, on the one hand, boosted the task of carrying out targeted and restricted brute-force attacks while, on the other, it has opened new and alternative authentication paths. Perhaps passwords have always been doomed since the very infancy of computer science, and now their usage is meeting its end imminently. Passwords used in web applications are broken. They are reused, phished and keylogged constantly by proliferating malware, or are leaked to the outside from time to time.

For the above reasons, efforts in this field are focused in finding novel and innovative passwordless authentication methods that overcome current drawbacks and that will keep up the pace with technology. Both the academia and the industry are pushing toward that objective which, in the meantime, will change security perceptions and awareness regarding authentication, starting off with the big players and ending with the end users. On the industry side, the main organizational bodies pioneering such efforts are the FIDO and the OATH. As described throughout Chapter 2, a few protocols are already under development, namely OSTP and OCRA. The combination of emerging technologies such as cloud computing, contactless NFC, smartphones and similar devices eases the process of making authentication transparent and seamless to users, while providing strong and mutual authentication by means of public-key cryptography, implemented throughout various factors of a MFA scheme. Authentication is converging not only to device-centric, but also to user-centric, as shown by the crumbs left throughout this dissertation. The TouchID technology delivered in the latest iPhone models is the most suitable

[1] Moore's Law was introduced by Intel co-founder Gordon Moore in 1965 [Moo98], saying that the number of transistors on integrated circuits doubles every two years.

example. Not only it is required to have the phone around for some 2FA via SMS or an offline application, but also the correct digital print of the thumb or any other finger for unlocking it. Despite its flaws—just as with any cutting-edge innovation—it is an optimistic leap toward the aforementioned objective.

The previous paragraph can be seen as a roadmap for authentication. The main obstacle is really the legacy. Passwords are so well integrated throughout IT that it is hard to let them go. Passwords also come in handy for backup schemes. In the second place, the FIDO and the OATH must also spread the word, invite and convince vendors to adhere the standards. Such task is long, effortful and time-consuming. Smartphones will eventually widely support NFC taps and biometric scanners to unlock the devices, decrypt offline data, and authenticate on cloud services by using such standards. It is also argued [PRSS13] that public-key cryptography based on cyclic groups modulo N is on a downhill. Lately, there has been breakthroughs on solving the DLP and on integer factorization that help devising efficient algorithms. This is highly concerning for the security of RSA and related public-key cryptography, and therefore it is also alarming for the security of Internet communications. In addition to the roadmaps outlined above, it is crucial to adopt new cryptographic standards in the long run. In spite of being a relatively new research field, ECC is notoriously a well-regarded cryptographic framework that can be used for all sorts of purposes, namely for authentication. However, though the efficiency and the number of tools for cryptanalysis is increasing, its real value in terms of practical usefulness is still low. The major concerns relate to the fundamental advances in mathematics which, despite coming only once in a while, can have a significant impact on widely deployed systems, says Bruce Schneier [Sch13b].

5.2 Main Conclusions

Science is undoubtedly the agent driving the evolution of technology. Fueling that evolution is the industry and the academia, both aiming at devising newer and better approaches for supporting a better and more secure cyberspace. The *cloud computing* technology surely resulted from that effort. As shown in Chapter 2, the *cloud* is today a catchword gaining attention from all over the academia, the industry and the media. It is sometimes used to refer to some service that some enterprise sells or offers, as means to abstract customers from the underlying details and allowing them to focus on their businesses. However, the true meaning of the word *cloud* may come from the virtualization technologies that propelled the cloud computing model to expand and be widely adopted in the form of XaaS. The commodity of paying-per-use with an on-demand self-serviced operation in replacement of classical IT is beneficial for the customers. Additionally, cloud computing vertically supports a wide spectrum in the basic forms of SaaS, PaaS, and IaaS, over a private, public, or hybrid deployment model. All of these models are suitable for enterprises and end users.

The advent of cloud computing has brought novel security issues specific to the technology and to how it is deployed over the Internet, as Chapter 3 explained, along with the problems posed by authentication approaches. Virtualization is a key element for the proliferation of IaaS clouds, but it has brought new security issues like cross-VM channels. The Internet cyberspace has been growing over the years fruitfully, but so as been the deep underground side of it in a transparent manner for most Internet users, which are not aware of the Internet dangers. Services on public clouds are, by default, susceptible to issues already present in the Internet and the technologies it uses. Particularly, cloud computing services require some interface providing capabilities to manage the subscribed service or services. Having the data on outsourced locations requires one to trust the entity in charge of the networking, storage and computation. Worst, the cloud is a shared environment accessed by other tenants. So, potentially sensitive data may be amongst other types of data, possibly unrelated, which belongs to other customers. The cloud computing model has further introduced more uncharted security risks. Given the importance this computing model already has these days and the prominence that most likely it will gain in the future, it is of interest for all to make it more secure. Until such risks are eliminated, potential cloud customers will yet remain reluctant about adopting cloud solutions. Because of less fortunate events from the past, the security awareness has started to take a more visible shape throughout standards, protocols, vendors, enterprises, and end users. There are efforts to mitigate the problems inherent to the cyberspace and technology of nowadays, and cloud computing is certainly within the scope of that effort.

The struggle to build stronger authentication methods is not new, but only now technology and vendors came together to support the proliferation of smartphones, QR codes, the BYOD paradigm, and consequently MFA and SSO combining a myriad of components to make authentication stronger and, eventually, give up password-based authentication. There are concerns around the placement of AD or LDAP servers together with password management. SSO comes to overcome this problem by centralizing authentication. But the concept lacks maturity, for which the main weakness is precisely the centralization. A single account may compromise several applications. MFA is largely appearing as 2FA in several online services, cloud-based or not, like Google and Facebook. In fact, banks already implemented a form of 2FA, even before the emergence of the MFA concept, so as to confirm online transactions based on SMS messages. And so the concept of OTP got materialized and is now regarded as an additional security barrier for authentication, requiring to have some software or hardware token around, or alternatively via SMS messages.

As described in Chapter 2, the research on this field has been focusing on providing new authentication factors, like the *someone you know*, or by using QR code technology to exchange cryptographic data in order for carrying out the execution of a particular protocol based on ZKP, ECC, or even based on passwords. On the other hand, the industry is leaning toward the

OSTP and OCRA that are specialized protocols that span entire hardware and software stacks while having in mind MFA, SSO, and strong and mutual authentication to cloud-based services. Risk-based authentication is also playing a small role in the field, but it requires further work to mature the concept and make it reliable since it is a heuristic approach. Moreover, biometric solutions like Authentify xFA portray a seamless experience, but the weakness comes with the unprotected smartphones vulnerable to malware. While the password is gradually disappearing, the new methods may also be vulnerable in some aspect. The aim is to make the Internet a safer place for all. Nonetheless, spreading the word is difficult for non-specialized people who are not aware of the security threats the Internet and the cyberspace presents. This huge flaw is largely exploited by a dark community motivated by the profits that may result from malicious activities. Social engineering tricks may be easy to spot for the expert who analyzes a spam email subjectively with basis on past experiences and knowledge. However, common Internet users or employees who visit email infrequently may be easily phished. It all boils down to the decision of clicking a link or opening a file. Nevertheless, in the wake of Edward Snowden and related NSA spying programs, the desire of spreading the words of security and privacy became more pronounced.

Enterprises are moving to the cloud, and the hybrid deployment model seems the most appropriate for balancing costs with trust, security, and compliance. The trend is clear. It is needed to adapt authentication mechanisms for the current networks and the current Internet state. To minimize the exposure gap imposed by the management interface of cloud computing services, this dissertation presented a robust model for protecting it, particularly while having in mind the authentication procedure. It is suitable for the hybrid cloud deployment model. This dissertation has studied the cloud computing model thoroughly while emphasizing the security issues it poses to the whole model, specifically to authentication. With this objective in mind, several approaches for authentication were described throughout this dissertation, explaining their assumptions, strengths and weaknesses, whenever possible, orthogonally to cloud computing. The discussions allowed to delineate the landscape of authentication methods and, consequently, a skeleton model for devising secure authentication frameworks for cloud environments, which is included in Chapter 4. The model recurs to an architecture based on a proxy gateway that mediates accesses to the inner web management interface. Each of these components are running on individual VMs, leveraging the benefits of cloud computing while providing higher security and resiliency against foreign threats. In the same chapter, a prototype for the model is described, showing how it can be deployed using modern technology, namely the Portuguese identity card, for achieving mutual authentication by using public-key cryptography with certificates. The Portuguese identity card embeds a powerful chip for performing cryptographic operations on-the-fly. With the standard firewall incorporated within Linux OSes, it was showed how a VM can be setup as a gateway proxy to communicate over a virtual network to a web server. The convenience of carrying two distinct certificates on the Portuguese identity

card also opens up possibilities to write applications that combine both, such as authenticating first and signing nonces afterward. The latter adheres to authentication in the long-term. For web sessions, it is important to make sure the one who initially authenticated is the one who is still in the session. Cryptographically bounding cookies to the underlying TLS channel seems like a good approach in this case.

Whether or not passwords are doomed to cease existence in the near future, for now several authentication approaches are still based on conventional passwords. Together with cloud computing and mobile computing, authentication will eventually become stronger by using innovative technologies that source biometric traits or utilize cryptography. The research field on cloud computing is also focused on patching the security issues and improve the overall security state of these environments. Eventually, security should be more spread throughout computer systems, particularly around cloud-based services with strong authentication in place.

5.3 Directions for Future Work

This dissertation described various approaches for authentication using different schemes combined with different technologies. Some of those approaches were merely possible because of the supporting technology. As a matter of a fact, technology will continue to evolve further, and thus it will have a more pronounced impact in terms of future authentication schemes, as explained in the first section of this chapter. In the next few years, the security of Internet communications and authentication will still rely on public-key cryptography. The advent of novel technology that underpins security is crucial for the creation of innovative authentication applications while utilizing well-known standards like RSA. The emergence of cloud computing has opened a door for developing innumerable applications. One of its applications has been showed in this dissertation. A model for authentication has been proposed, and a PoC showing its applicability was implemented. Both are prone to further work.

The proxy gateway is a critical centralized point for the security of the model. It was said that security controls can be installed on that point with little effort, in addition to the firewall. As such, more practical restrictions and virtual appliances can be added to monitor particular security components, and thus increase the proxy gateway to make it more robust. In essence, it can function like an amalgamation of particular resources useful for authentication, being able to construct security profiles and deliver them as a service. Each profile would vary according to the level of security desired and the associated overhead. To study how the security resources are combined and the outcome of such combination is a possible direction for future work. The usefulness of employing various SVAs would be assessed.

To test various instances of the model under atypical conditions is also left as future work. In other words, the implementation of the model, like the prototype using the Portuguese identity

card, is theoretically more resistant against a number of outsider and insider attacks. Testing several implementations against the efficiency and overhead under abnormal conditions, such as a packet flood, would be useful to demonstrate the resiliency of the VM as well as the delay in automatically booting up new VMs to aid mitigating the attack. Quantify how much such an operation would cost and aid in terms of security are further directions for future work as well.

References

[11] News: RSA hack leaves status of SecurID uncertain. *Netw. Secur.*, 2011(4):1–2, Apr. 2011. xviii, 52

[57u13] 57un Blog. A BIG password cracking wordlist. Available in `https://57un.wordpress.com/2013/03/09/a-big-password-cracking-wordlist/`, Mar. 2013. Accessed Apr. 2013. xviii, 47

[AALX+12] H. Al-Aqrabi, Lu Liu, Jie Xu, R. Hill, N. Antonopoulos, and Yongzhao Zhan. Investigation of IT Security and Compliance Challenges in Security-as-a-Service for Cloud Computing. In *Proc. of the 15th IEEE Int. Symp. on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops (ISORCW)*, pages 124–129, Shenzhen, Guangdong, China, Apr. 2012. 12

[AFG+10] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. A View of Cloud Computing. *Commun. ACM*, 53(4):50–58, 2010. 42, 43

[Air] Aircrack-ng. Aircrack-ng Website. Available in `http://www.aircrack-ng.org/`. Accessed Aug. 2013. 47

[Ama] Amazon. Amazon VPC Website. Available in `https://aws.amazon.com/vpc/`. Accessed Jul. 2013. 15

[Ama12] Amazon. Amazon Elastic Compute Cloud (Amazon EC2). Available in `https://aws.amazon.com/ec2/`, 2012. Accessed Apr. 2013. 13

[Ass13] Associated Press. Hackers compromise AP Twitter account. Available in `http://bigstory.ap.org/article/hackers-compromise-ap-twitter-account`, Apr. 2013. Accessed May 2013. 52

[Aut] Authentify. xFA. Available in `http://www.authentify.com/xFA/`. Accessed Sep. 2013. xx, 36, 69

[Bal12a] Dirk Balfanz. Channel-Bound Cookies. Available in `http://www.browserauth.net/channel-bound-cookies`, 2012. Accessed May 2013. 22

[Bal12b] Dirk Balfanz. [TLS] Update on Origin-Bound Certificates: Now called "Channel ID". Available in `https://www.ietf.org/mail-archive/web/tls/current/msg09042.html`, Nov. 2012. Accessed Sep. 2013. 22

[BBBB10] Hristo Bojinov, Elie Bursztein, Xavier Boyen, and Dan Boneh. Kamouflage: Loss-Resistant Password Management. In *Proc. of the 15th European Symp. on Research in Computer Security (ESORICS)*, pages 286–302, Athens, Greece, Sep. 2010. Springer-Verlag. 20

[BH12] Dirk Balfanz and Ryan Hamilton. Transport Layer Security (TLS) Channel IDs. Available in https://tools.ietf.org/html/draft-balfanz-tls-channelid-00, Nov. 2012. xvii, 21

[BJR+06] John Brainard, Ari Juels, Ronald L. Rivest, Michael Szydlo, and Moti Yung. Fourth-Factor Authentication: Somebody You Know. In *Proc. of the 13th ACM Conf. on Computer and Communications Security (CCS)*, pages 168–178, Alexandria, VA, USA, Oct.-Nov. 2006. ACM. 28, 35

[BNP+11] Sven Bugiel, Stefan Nürnberger, Thomas Pöppelmann, Ahmad-Reza Sadeghi, and Thomas Schneider. AmazonIA: When Elasticity Snaps Back. In *Proc. of the 18th ACM Conf. on Computer and Communications Security*, pages 389–400, New York, NY, USA, Oct. 2011. ACM. 42

[BW12] Philogene A. Boampong and Luay A. Wahsheh. Different Facets of Security in the Cloud. In *Proc. of the 15th Communications and Networking Simulation Symp.*, pages 5:1–5:7, San Diego, CA, USA, 2012. Society for Computer Simulation International. 42

[Cha13] Chaos Computer Club. Chaos Computer Club breaks Apple TouchID. Available in http://www.ccc.de/en/updates/2013/ccc-breaks-apple-touchid, Sep. 2013. Accessed Sep. 2013. 50

[Cis13] Cisco. 2013 Cisco Annual Security Report. Available in http://www.cisco.com/en/US/prod/vpndevc/annual_security_report.html, Mar. 2013. Accessed Apr. 2013. 15, 51

[CKS+11] A.J. Choudhury, P. Kumar, M. Sain, Hyotaek Lim, and Hoon Jae-Lee. A Strong User Authentication Framework for Cloud Computing. In *Proc. of the IEEE Asia-Pacific Services Computing Conference (APSCC)*, pages 110–115, Jeju, South Korea, Dec. 2011. 33

[CLJ+11] Kyeongwon Choi, Changbin Lee, Woongryul Jeon, Kwangwoo Lee, and Dongho Won. A mobile based Anti-Phishing Authentication Scheme using QR code. In *Int. Conf. on Mobile IT Convergence (ICMIC)*, pages 109–113, Gumi, South Korea, Sep. 2011. xiv, xvii, 33

[ClYS11] Tien-Ho Chen, Hsiu lien Yeh, and Wei-Kuan Shih. An Advanced ECC Dynamic ID-Based Remote Mutual Authentication Scheme for Cloud Computing. In *Proc. of the 5th FTRA Int. Conf. on Multimedia and Ubiquitous Engineering (MUE)*, pages 155–159, Loutraki, Greece, Jun. 2011. 34

[CPK10] Yanpei Chen, Vern Paxson, and Randy H. Katz. What's New About Cloud Computing Security? Technical Report UCB/EECS-2010-5, EECS Department, University of California, Berkeley, 2010. Available from: http://www.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-5.html. 44

[Cur13] Sam Curry. Transforming Identity Assurance through Risk-Based Authentication. Available in http://pulseblog.emc.com/2013/02/26/transforming-identity-assurance-through-risk-based-authentication-2/, Feb. 2013. Accessed Jul. 2013. 34

[Dav13] Dave Engberg. Service-wide Password Reset. Available in http://blog.evernote.com/blog/2013/03/02/security-notice-service-wide-password-reset/comment-page-2/, Mar. 2013. Accessed Apr. 2013. 49

[Daz] Dazzlepod. Disclosure Project: UNIQPASS. Available in https://dazzlepod.com/uniqpass/. Accessed May 2013. xiv, 47

[Doc13] Doctor Web. New Trojan steals short messages. Available in http://news.drweb.com/show/?i=3549, May 2013. Accessed May 2013. 51

[Doe12] Kevin Doel. Scary Logins: Worst Passwords of 2012 and How to Fix Them. Available in http://splashdata.com/press/pr121023.htm, 2012. 45

[Ele] Electronic Frontier Foundation. HTTPS Everywhere Website. Available in https://www.eff.org/https-everywhere. Accessed Apr. 2013. 70

[ENI09] ENISA. Cloud Computing: Benefits, Risks and Recommendations for Information Security. Available in http://www.enisa.europa.eu/activities/risk-management/files/deliverables/cloud-computing-risk-assessment, Nov. 2009. Accessed Sep. 2012. 43

[FID] FIDO. FIDO Website. Available in http://www.fidoalliance.org/. Accessed May 2013. xvii, 23, 30

[Fir13] FireEye. FireEye Advanced Threat Report - 2H 2012. Available in http://www2.fireeye.com/rs/fireye/images/fireeye-advanced-threat-report-2h2012.pdf, Mar. 2013. Accessed Apr. 2013. 51

[Fon13] John Fontana. Two-factor authentication in two years. ZDNet, Apr. 2013. 49

[Fri13] Lex Friedman. Apple adds two-step authentication option for iCloud, Apple IDs. Macworld, Mar. 2013. 49

[FSFI13] Diogo A. B. Fernandes, Liliana F. B. Soares, Mário M. Freire, and Pedro R. M. Inácio. Randomness in Virtual Machines. In *Proc. of the 6th IEEE/ACM Int. Conf. on Utility and Cloud Computing (UCC)*, Dresden, Germany, Dec. 2013. IEEE Computer Society. Accepted for publication. 42

[FSG+13a] Diogo A. B. Fernandes, Liliana F. B. Soares, João V. Gomes, Mário M. Freire, and Pedro R. M. Inácio. A Quick Perspective on the Current State in Cybersecurity. In Babak Akhgar and Hamid R. Arabnia, editors, *Emerging Trends in Information and Communication Technologias Security*, pages 423–441. Elsevier (Morgan Kaufmann), 2013. In press. xvi, 6, 17, 39

[FSG+13b] Diogo A. B. Fernandes, Liliana F. B. Soares, João V. Gomes, Mário M. Freire, and Pedro R. M. Inácio. Security Issues in Cloud Environments — A Survey. *Int. J. Inf. Secur.: Security in Cloud Computing*, 2013. Available from: http://link.springer.com/article/10.1007%2Fs10207-013-0208-7. xvi, xxix, 5, 11, 15, 17, 39, 40

[Gaz] Gazzang. zTrustee. Available in http://www.gazzang.com/products/ztrustee. Accessed Aug. 2013. 35

[GCC11] S. Grzonkowski, P.M. Corcoran, and T. Coughlin. Security Analysis of Authentication Protocols for Next-Generation Mobile and CE Cloud Services. In *Proc. of the IEEE Int. Conf. on Consumer Electronics - Berlin (ICCE-Berlin)*, pages 83–87, Berlin, Germany, Sep. 2011. 34

[GMR+11] N. Gonzalez, C. Miers, F. Redigolo, T. Carvalho, M. Simplicio, M. Naslund, and M. Pourzandi. A Quantitative Analysis of Current Security Concerns and Solutions for Cloud Computing. In *Proc. of the IEEE 3rd Int. Conf. on Cloud Computing Technology and Science*, pages 231–238, Washington, D.C., USA, Nov.–Dec. 2011. IEEE Computer Society. 43

[Gooa] Google. AuthSub for Web Applications. Available in https://developers.google.com/accounts/docs/AuthSub. Accessed Sep. 2013. 26

[Goob] Google. Google App Engine Website. Available in https://developers.google.com/appengine/. Accessed Apr. 2013. 12

[Gooc] Google. Google Authenticator. Available in https://code.google.com/p/google-authenticator/. Accessed Jul. 2013. xx, 37

[Goo12] Dan Goodin. Why passwords have never been weaker—and crackers have never been stronger. Ars Technica, Aug. 2012. xiv, xviii, 47

[GPa06] GPayments. Two-Factor Authentication: An essential guide in the fight against Internet fraud. Available in http://www.gpayments.com/pdfs/WHITEPAPER_2FA-Fighting_Internet_Fraud.pdf, Feb. 2006. Accessed May 2013. 27

[Gre13] Andy Greenberg. iOS 7 Bug Lets Anyone Bypass iPhone's Lockscreen To Hijack Photos, Email, Or Twitter. Available in http://www.forbes.com/sites/andygreenberg/2013/09/19/ios-7-bug-lets-anyone-bypass-iphones-lockscreen-to-hijack-photos-email-or-twitter/, Sep. 2013. Accessed Sep. 2013. xix, 50

[GU13] Eric Grosse and Mayank Upadhyay. Authentication at Scale. *IEEE Secur. Privacy*, 11(1):15–22, 2013. xiv, xvii, 26

[Hah] Hahscat. Hashcat Website. Available in http://hashcat.net/. Accessed Aug. 2013. 47

[HAKH10] H. Haleh, H. Akbrzade Khorshidi, and S. M. Hoseini. A new approach for fuzzy risk analysis based on similarity by using Decision Making Approach. In *Proc. of the IEEE Int. Conf. on Management of Innovation and Technology (ICMIT)*, pages 1112–1117, Singapore, Jun. 2010. 34

[Hal10] Coda Hale. bcrypt. Available in `http://codahale.com/how-to-safely-store-a-password/`, 2010. Accessed May 2013. xx, 70

[Hon12] Mat Honan. How Apple and Amazon Security Flaws Led to My Epic Hacking. Wired, Aug. 2012. 49

[Hyp13] Mikko Hypponnen. Twitter's 2FA: SMS Double-Duty. Available in `http://www.f-secure.com/weblog/archives/00002560.html`, May 2013. Accessed May 2013. 52

[HZ11] Yu N. Hao Z., Zhong S. A Time-Bound Ticket-Based Mutual Authentication Scheme for Cloud Computing. *International Journal of Computers Communications & Control*, 6(2):227–235, Jun. 2011. 34

[Imp10] Imperva. Consumer Password Worst Practices. Available in `http://www.imperva.com/docs/WP_Consumer_Password_Worst_Practices.pdf`, 2010. Accessed May 2013. 45

[Inf] Information is Beautiful. World's Biggest Data Breaches. Available in `http://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/`. Accessed Aug. 2013. 46

[Ins] InsidePro. Extreme GPU Bruteforcer Website. Available in `http://www.insidepro.com/eng/egb.shtml`. Accessed Aug. 2013. 47

[Jai13] C.D. Jaidhar. Enhanced Mutual Authentication Scheme for Cloud Architecture. In *Proc. of the IEEE 3rd Int. Advance Computing Conference (IACC)*, pages 70–75, Ghaziabad, India, Feb. 2013. 34

[JB08] Collin Jackson and Adam Barth. ForceHTTPS: Protecting High-Security Web Sites from Network Attacks. In *Proc. of the 17th Int. Conf. on World Wide Web (WWW)*, pages 525–534, Beijing, China, Apr. 2008. ACM. 70

[JGH09] Meiko Jensen, Nils Gruschka, and Ralph Herkenhöner. A survey of attacks on web services. *Computer Science - Research and Development*, 24:185–197, 2009. 10.1007/s00450-009-0092-6. xviii, 43

[Jim13] Jimio. Improvements to login verification, photos and more. Available in `https://blog.twitter.com/2013/improvements-to-login-verification-photos-and-more`, Aug. 2013. Accessed Aug. 2013. 53

[JR13] Ari Juels and Ronald L. Rivest. Honeywords: Making Password-Cracking Detectable. Available in http://people.csail.mit.edu/rivest/pubs/JR13.pdf, May 2013. Version 2. Accessed May 2013. xvii, 20

[KBC97] H. Krawczyk, M. Bellare, and R. Canetti. HMAC: Keyed-Hashing for Message Authentication. Available in http://www.ietf.org/rfc/rfc2104.txt, February 1997. Updated by RFC 6151. 29

[Ker13a] Brian Kerbs. Adobe To Announce Source Code, Customer Data Breach. Available in https://krebsonsecurity.com/2013/10/adobe-to-announce-source-code-customer-data-breach/, Sep. 2013. xiv, xviii, 46

[Ker13b] Brian Kerbs. DDoS Services Advertise Openly, Take PayPal. Available in https://krebsonsecurity.com/2013/05/ddos-services-advertise-openly-take-paypal/, May. 2013. Accessed Aug. 2013. 40

[Kir13] Dustin Kirkland. Multi-factor Authentication in the Cloud. Available in https://events.linuxfoundation.org/images/stories/slides/lfcs2013_kirkland.pdf, Apr. 2013. Accessed May 2013. 25, 26

[KKM+12] P.G. Kelley, S. Komanduri, M.L. Mazurek, R. Shay, T. Vidas, L. Bauer, N. Christin, L.F. Cranor, and J. Lopez. Guess Again (and Again and Again): Measuring Password Strength by Simulating Password-Cracking Algorithms. In *Proc. of the IEEE Symp. on Security and Privacy (SP)*, pages 523–537, San Francisco, CA, USA, May 2012. xiv, xviii, 47

[KLM+10] Peter Kieseberg, Manuel Leithner, Martin Mulazzani, Lindsay Munroe, Sebastian Schrittwieser, Mayank Sinha, and Edgar Weippl. QR Code Security. In *Proc. of the 8th Int. Conf. on Advances in Mobile Computing and Multimedia (MoMM)*, pages 430–435, Paris, France, 2010. ACM. xviii, 50

[KW13] Dhiru Kholia and Przemysław Węgrzyn. Looking inside the (Drop) box. In *7th USENIX Workshop on Offensive Technologies (WOOT)*, pages 1–7, Washington, DC, USA, Aug. 2013. 53

[Lai08] Dao Yu Lai. Intelligent Online Risk-Based Authentication using Bayesian Network Model. Master's thesis, University of Victoria, 2008. 34

[Las] LastPass. LastPass Website. Available in https://lastpass.com/. Accessed May 2013. 20

[Lig] Lightning Wire Labs. IPFire Website. Available in http://www.ipfire.org/. Accessed Oct. 2013. 61

[Liv13] LivingSocial. LivingSocial security notice. Available in https://livingsocial.com/createpassword, May 2013. Accessed May 2013. xviii, 46

[LKL+10] Young Sil Lee, Nack Hyun Kim, Hyotaek Lim, HeungKuk Jo, and Hoon-Jae Lee. Online Banking Authentication System using Mobile-OTP with QR-code. In *Proc. of the 5th Int. Conf. on Computer Sciences and Convergence Information Technology (ICCIT)*, pages 644–648, Dhaka, Bangladesh, Dec. 2010. 33

[LL10] Kuan-Chieh Liao and Wei-Hsun Lee. A Novel User Authentication Scheme Based on QR-Code. *Journal of Networks*, 5(8):1–5, Aug. 2010. 32

[MA11] S. Mukhopadhyay and D. Argles. An Anti-Phishing mechanism for Single Sign-On based on QR-code. In *Int. Conf. on Information Society (i-Society)*, pages 505–508, London, United Kingdom, Jun. 2011. 33

[Mao13] Etay Maor. Perfectionism, Fraudster Style. Trusteer Blog, Apr. 2013. 51

[Mar13] D. Martin. Implementing Effective Controls in a Mobile, Agile, Cloud-Enabled Enterprise. *IEEE Secur. Privacy*, 11(1):13–14, 2013. xx, 70

[MBH+05] D. M'Raihi, M. Bellare, F. Hoornaert, D. Naccache, and O. Ranen. HOTP: An HMAC-Based One-Time Password Algorithm. Available in `http://www.ietf.org/rfc/rfc4226.txt`, Dec. 2005. 29

[McA13a] McAfee. McAfee Cloud Single Sign On. Available in `http://www.mcafee.com/us/products/cloud-single-sign-on.aspx`, 2013. Accessed Aug. 2013. xvii, 25

[McA13b] McAfee. McAfee Threats Report - Fourth Quarter 2012. Available in `http://www.mcafee.com/us/resources/reports/rp-quarterly-threat-q4-2012.pdf`, Apr. 2013. Accessed Apr. 2013. 50

[MFH11] Ziqing Mao, Dinei A. F. Florêncio, and Cormac Herley. Painless Migration from Passwords to Two Factor Authentication. In *Proc. of the IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE, Nov. 2011. 69

[Mim13] Michael Mimoso. Vulnerability in Viber for Android Enables Lock Screen Bypass. Available in `https://threatpost.com/vulnerability-in-viber-for-android-enables-lock-screen-bypass`, Apr. 2013. 50

[MMPR11] D. M'Raihi, S. Machani, M. Pei, and J. Rydell. TOTP: Time-Based One-Time Password Algorithm. Available in `http://www.ietf.org/rfc/rfc6238.txt`, May 2011. 29

[Moo98] Gordon E. Moore. Cramming More Components Onto Integrated Circuits. *Proc. of the IEEE*, 86(1):82–85, 1998. 73

[Moz] Mozilla Foundation. BrowserID. Available in `https://github.com/mozilla/id-specs/blob/prod/browserid/index.md`. Accessed Aug. 2013. 25

[MRB+11] D. M'Raihi, J. Rydell, S. Bajaj, S. Machani, and D. Naccache. OCRA: OATH Challenge-Response Algorithm. Available in `http://www.ietf.org/rfc/rfc6287.txt`, Jun. 2011. 30

[Nar12] Ryan Naraine. Google testing login authentication via QR codes. Available in http://www.zdnet.com/two-factor-authentication-in-two-years-7000013474/, Jan. 2012. Accessed on Jun. 2013. 33, 36

[NIS11] NIST. The NIST Definition of Cloud Computing. Available in http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf, Sept. 2011. Accessed Sep. 2012. 10

[OAT] OATH. OATH Website. Available in http://www.openauthentication.org/. Accessed May 2013. xvii, 23

[OAu] OAuth. OAuth Website. Available in http://oauth.net/. Accessed Sep. 2013. 26

[OCJ08] Jon Oberheide, Evan Cooke, and Farnam Jahanian. Empirical Exploitation of Live Virtual Machine Migration. In *Proc. of the Black Hat Conv.*, Aug. 2008. 41

[Ope] Openwall. John the Ripper Website. Available in http://www.openwall.com/john/. Accessed Aug. 2013. 47

[Ope07] OpenID Foundation. OpenID Authentication 2.0 Specification. Available in https://openid.net/specs/openid-authentication-2_0.txt, 2007. Accessed Aug. 2013. 25

[Oph] Ophcrack. Ophcrack Website. Available in http://ophcrack.sourceforge.net/. Accessed Aug. 2013. 47

[Ora13] Oracle. VirtualBox Website. Available in https://www.virtualbox.org/, 2013. Accessed Jun. 2013. 13

[Ort12] Alberto Ortega. Your malware shall not fool us with those anti analysis tricks. AlienVault Labs, Nov. 2012. 42

[PA12] Peirluigi Paganini and Richard Amores. *The Deep Dark Web: the hidden world*, volume 1. CreateSpace, 2012. xvii, 40

[Pag13] Pierluigi Paganini. Cybercrime as a Service. Available in http://resources.infosecinstitute.com/cybercrime-as-a-service/, Aug. 2013. Accessed Aug. 2013. 40

[Par13] Parallels. Parallels Website. Available in http://www.parallels.com/eu/products/, 2013. Accessed Jun. 2013. 13

[PCG+13] Luís Perez, Jason Costa, João Gomes, Mário Freire, and Pedro Inácio. Strong Authentication with Quick Response Codes. In *Proc. of the 9th Conf. on Telecommunications (ConfTele2013)*, Castelo Branco, Portugal, May 2013. 33

[Pea13] Siani Pearson. Privacy, Security and Trust in Cloud Computing. In *Privacy and Security for Cloud Computing*, pages 3–42. Springer London, 2013. xiv, xviii, 43

[Pon13] Ponemon Institute. Moving Beyond Passwords: Consumer Attitudos on Online Authentication. Available in http://go.noknok.com/rs/noknok/images/

NokNok-Ponemon-ExecutiveSummary-Apr13.pdf, Apr. 2013. Accessed May 2013. 46

[Pora] Portugal Telecom. SmartCloudPT Website. Available in http://www.smartcloudpt.pt/. Accessed Aug. 2013. 12

[Porb] Portuguese Government. Portuguese Identity Card PKI Website. Available in https://pki.cartaodecidadao.pt/. Accessed Sep. 2013. 62, 67

[Porc] Portuguese Government. Portuguese Identity Card SDK Website. Available in http://www.kitcc.pt/ccidadao/kits. Accessed Sep. 2013. 62

[Pord] Portuguese Government. Portuguese Identity Card Website. Available in http://www.cartaodecidadao.pt/index.php%3Flang=en.html. Accessed Sep. 2013. 62, 63

[PRSS13] Thomas Ptacek, Tom Ritter, Javed Samuel, and Alex Stamos. The Factoring Dead: Preparing for the Cryptopocalypse. Available in https://www.blackhat.com/us-13/archives.html#Stamos, Aug. 2013. Accessed Aug. 2013. 74

[Pu10] Qiong Pu. An Improved Two-factor Authentication Protocol. In *Proc. of the Second Int. Conf. on Multimedia and Information Technology (MMIT)*, volume 2, pages 223–226, Kaifeng, China, Apr. 2010. 33

[RC11] Francisco Rocha and Miguel Correia. Lucy in the Sky without Diamonds: Stealing Confidential Data in the Cloud. In *Proc. of the IEEE/IFIP 41st Int. Conf. on Dependable Systems and Networks Workshops*, pages 129–134. IEEE, Jun. 2011. xviii, 42

[Rei] Dominik Reichl. KeePass Password Safe. Available in http://keepass.info/. Accessed Sep. 2013. xvii, 20

[Rep13] RepoCERT. Botnet using Plesk vulnerability and takedown. Available in http://seclists.org/fulldisclosure/2013/Jun/36, Jun. 2013. Accessed Jul. 2013. 44

[RMVC+12] Luis Rodero-Merino, Luis M. Vaquero, Eddy Caron, Fréderic Desprez, and Aarian Muresan. Building Safe PaaS clouds: a Survey on Security in Multitenant Software Platforms. *Computers & Security*, 31(1):96–108, Jan. 2012. 12, 42

[RQSL12] Oriana Riva, Chuan Qin, Karin Strauss, and Dimitrios Lymberopoulos. Progressive authentication: deciding when to authenticate on mobile phones. In *Proc. of the 21st USENIX Conf. on Security Symp.*, Security'12, pages 1–16, Bellevue, WA, USA, 2012. USENIX Association. xvii, 34

[RSAa] RSA. RSA Authentication Manager. Available in http://www.emc.com/security/rsa-securid/rsa-authentication-manager.htm. Accessed Aug. 2013. xvii, 26

[RSAb] RSA Laboratories. PKCS #11: Cryptographic Token Interface Standard. Available in http://www.emc.com/emc-plus/rsa-labs/standards-initiatives/pkcs-11-cryptographic-token-interface-standard.htm. Accessed Sep. 2013. 63

[RSAc] RSA Security. RSA SecurID. Available in http://www.emc.com/security/rsa-securid.htm. Accessed Aug. 2013. 35

[RTSS09] Thomas Ristenpart, Eran Tromer, Hovav Shacham, and Stefan Savage. Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds. In *Proc. of the 16th ACM Conf. on Computer and Communications Security*, pages 199–212, Chicago, IL, USA, Nov. 2009. ACM. xiv, xviii, 42

[Rya12] Konstantin Ryabitsev. Limitations of Google Authenticator pam module. Available in http://blog.mricon.com/2012/12/limitations-of-google-authenticator-pam.html, Dec. 2012. Accessed Sep. 2013. 37, 70

[SACZ+13] K. Salah, J.M. Alcaraz Calero, S. Zeadally, S. Al-Mulla, and M. Alzaabi. Using Cloud Computing to Implement a Security Overlay Network. *IEEE Secur. Privacy*, 11(1):44–53, 2013. xix, 58

[SAM05] SAML v2.0. OASIS Website. Available in https://www.oasis-open.org/standards#samlv2.0, 2005. Accessed Apr. 2013. 24

[Sch05] Bruce Schneier. Two-Factor Authentication: Too Little, Too Late. *Commun. ACM*, 48(4):136–136, Apr. 2005. 48

[Sch13a] Bruce Schneier. NSA Storing Internet Data, Social Networking Data, on Pretty Much Everybody. Available in https://www.schneier.com/blog/archives/2013/10/nsa_storing_int.html, Oct. 2013. Accessed Oct. 2013. xviii, 40

[Sch13b] Bruce Schneier. The Cryptopocalypse. Available in https://www.schneier.com/blog/archives/2013/08/the_cryptopocal.html, Aug. 2013. Accessed Aug. 2013. xxi, 74

[Sch13c] Mathew J. Schwartz. Zeus Bank Malware Surges On Facebook. Available in https://www.informationweek.com/security/attacks/zeus-bank-malware-surges-on-facebook/240156156, Jun. 2013. 49

[SE13] A.K. Sood and R.J. Enbody. Targeted Cyberattacks: A Superset of Advanced Persistent Threats. *IEEE Secur. Privacy*, 11(1):54–61, 2013. xiii, 40, 44

[Sel13] Stefan Sellmer. No paysafecard needed, your passwords will pay off. Available in https://blogs.technet.com/b/mmpc/archive/2013/05/16/no-paysafecard-needed-your-passwords-will-pay-off.aspx?Redirected=true, May. 2013. Accessed May 2013. 50

[SFFI13] Liliana F. B. Soares, Diogo A. B. Fernandes, Mário M. Freire, and Pedro R. M. Inácio. Secure User Authentication in Cloud Computing Management Interfaces. In *Proc. of the 32nd IEEE International Performance Computing and Communications Conference (IPCCC)*, San Diego, CA, USA, Dec. 2013. IEEE Computer Society. Accepted for publication. xvi, 6, 40

[SFG+14] Liliana F. B. Soares, Diogo A. B. Fernandes, João V. Gomes, Mário M. Freire, and Pedro R. M. Inácio. Cloud Security: State of the Art. In Surya Nepal and Mukaddim Pathan, editors, *Security, Privacy and Trust in Cloud Systems*. Springer, Berlin Heidelberg, 2014. In press. xvi, 5, 11, 39, 55

[Sha85] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Proc. of the CRYPTO 84 on Advances in Cryptology*, pages 47–53, Santa Barbara, CA, USA, 1985. Springer-Verlag New York, Inc. 23

[Sli13] Eric Slivka. iOS 6.1 Bug Enables Bypassing Passcode Lock to Access Phone and Contacts. Available in http://www.macrumors.com/2013/02/14/ios-6-1-bug-enables-bypassing-passcode-lock-to-access-phone-and-contacts/, Feb. 2013. 50

[Str13] David Strom. Integrating Single Sign-on Across the Cloud. Available in http://www.webtorials.com/content/2013/08/integrating-single-sign-on-across-the-cloud.html, 2013. Accessed Sep. 2013. 58

[Tay] Steven Taylor. CAPTCHA on its Last Legs for Authentication. Available in http://www.webtorials.com/content/2013/10/captcha-on-its-last-legs-for-authentication.html. Accessed Oct. 2013. 46

[The13] The Linux Foundation. Xen Website. Available in http://http://www.xenproject.org/, 2013. Accessed Jun. 2013. 13

[Tho] Thoughtcrime Labs. CloudCracker Website. Available in https://cloudcracker.com/. Accessed Aug. 2013. 47

[Tho13] H. Thompson. The Human Element of Information Security. *IEEE Secur. Privacy*, 11(1):32–35, 2013. 40

[Tow09] Mark Townsend. Managing a Security Program in a Cloud Computing Environment. In *Information Security Curriculum Development Conf.*, pages 128–133, New York, NY, USA, 2009. ACM. xviii, 45

[Tra12] Tiffany Trader. GPU Monster Shreds Password Hashes. Available in http://www.hpcwire.com/hpcwire/2012-12-06/gpu_monster_shreds_password_hashes.html, Dec. 2012. 47

[Tro13] Troy Hunt. 5 ways to implement HTTPS in an insufficient manner (and leak sensitive data). Available in http://www.troyhunt.com/2013/04/5-ways-to-implement-https-in.html, Apr. 2013. Accessed Apr. 2013. xviii, 43, 59

[Tru13] Trustwave. Trustwave 2013 Global Security Report. Available in https://www2.trustwave.com/2013GSR.html, 2013. Accessed May 2013. xviii, 45

[TWO+12] I. Traore, I. Woungang, M.S. Obaidat, Y. Nakkabi, and I. Lai. Combining mouse and keystroke dynamics biometrics for risk-based authentication in web environments. In *Proc. of the 4th Int. Conf. on Digital Home (ICDH)*, pages 138–145, Guangzhou, China, Nov. 2012. xiv, 34

[VCB+13] G.S. Veronese, M. Correia, A.N. Bessani, Lau Cheuk Lung, and P. Verissimo. Efficient Byzantine Fault-Tolerance. *EEE Transactions on Computers*, 62(1):16–30, 2013. 60

[Ver12] Versafe and CheckPoint. A Case Study of Eurograbber. Available in `https://www.checkpoint.com/products/downloads/whitepapers/Eurograbber_White_Paper.pdf`, Dec. 2012. Accessed Jan. 2013. 51

[VMw13] VMware. VMware Website. Available in `https://www.vmware.com/products/`, 2013. Accessed Jun. 2013. 13

[VPT+12] Quang Hieu Vu, Tran-Vu Pham, Hong-Linh Truong, S. Dustdar, and R. Asal. DEMODS: A Description Model for Data-as-a-Service. In *Proc. of the IEEE 26th Int. Conf. on Advanced Information Networking and Applications (AINA)*, pages 605–612, Fukuoka, Japan, Mar. 2012. 12

[VRMM11] Luis M. Vaquero, Luis Rodero-Merino, and Daniel Morán. Locking the sky: a survey on IaaS cloud security. *Computing*, 91(1):93–118, Jan. 2011. 41

[Web13] Web Sense. WebSense 2013 Threat Report. Available in `https://www.websense.com/assets/reports/websense-2013-threat-report.pdf`, Mar. 2013. Accessed Apr. 2013. xviii, 51

[Wei07] Aaron Weiss. Computing in the Clouds. *netWorker*, 11(4):16–25, 2007. 9

[Who] Whonix Operating System. Whonix Website. Available in `https://www.whonix.org/wiki/Main_Page`. Accessed Sep. 2013. xix, 56

[WZA+09] Jinpeng Wei, Xiaolan Zhang, Glenn Ammons, Vasanth Bala, and Peng Ning. Managing Security of Virtual Machine Images in a Cloud Environment. In *Proc. of the ACM Workshop on Cloud Computing Security*, pages 91–96, New York, NY, USA, Nov. 2009. ACM. 41

[XX13] Zhifeng Xiao and Yang Xiao. Security and Privacy in Cloud Computing. *IEEE Commun. Surveys Tuts.*, 15(2):843–859, 2013. xiv, xviii, 41, 42, 43

[XXHW13] Jidong Xiao, Zhang Xu, Hai Huang, and Haining Wang. Security Implications of Memory Deduplication in a Virtualized Environment. In *Proc. of the 43rd Annual IEEE/IFIP Int. Conf. on Dependable Systems and Networks (DSN)*, pages 1–12, Budapest, Hungary, Jun. 2013. 42

[Yah] Yahoo. Browser-Based Authentication. Available in `http://developer.yahoo.com/bbauth/`. Accessed Sep. 2013. 26

[YC09] Jen-Ho Yang and Chin-Chen Chang. An ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem. *Computers & Security*, 28(3–4):138–143, 2009. 34

[YJI+12] A. Ali Yassin, Hai Jin, A. Ibrahim, Weizhong Qiang, and Deqing Zou. A Practical Privacy-preserving Password Authentication Scheme for Cloud Computing. In *Proc. of the IEEE 26th Int. Parallel and Distributed Processing Symp. Workshops PhD Forum (IPDPSW)*, pages 1210–1217, Phoenix, AZ, USA, May 2012. 34

[YJIZ12] A. Ali Yassin, Hai Jin, A. Ibrahim, and Deqing Zou. Anonymous Password Authentication Scheme by Using Digital Signature and Fingerprint in Cloud Computing. In *Proc. of the Second Int. Conf. on Cloud and Green Computing (CGC)*, pages 282–289, Xiangtan, Hunan, China, Nov. 2012. xiv, 33, 49

[Yub] Yubico. YubiKey Hardware. Available in `https://www.yubico.com/products/yubikey-hardware/`. Accessed May 2013. 36

[YY09] Eun-Jun Yoon and Kee-Young Yoo. Robust ID-Based Remote Mutual Authentication with Key Agreement Scheme for Mobile Devices on ECC. In *Int. Conf. on Computational Science and Engineering (CSE)*, volume 2, pages 633–640, Vancouver, Canada, Aug. 2009. 34

[Zar] Eugene Zarakhovsky. Summary of the August 13th App Outage. Available in `https://developers.facebook.com/blog/post/2013/08/15/summary-of-the-august-13th-app-outage/`. Accessed Sep. 2013. 49

[ZJRR12] Yinqian Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. Cross-VM Side Channels and Their Use to Extract Private Keys. In *Proc. of the 19th ACM Conf. on Computer and Communications Security (CCS)*, pages 305–316, Raleigh, NC, USA, Oct. 2012. ACM. 42

[ZZX+10] Minqi Zhou, Rong Zhang, Wei Xie, Weining Qian, and Aoying Zhou. Security and Privacy in Cloud Computing: A Survey. In *Proc. of the 6th Int. Conf. on Semantics Knowledge and Grid*, pages 105–112, Washington, D.C., USA, Nov. 2010. IEEE Computer Society. xiv, xviii, 42, 43