e-certsDS: Certificados Eletrônicos com Assinatura Digital

Maurício El Uri[1], **Luciano Vargas**[1], **Diego Kreutz**[1]

[1]Universidade Federal do Pampa (Unipampa)

mauricioeluri@gmail.com,{lucianovargas,diegokreutz}@unipampa.edu.br

Resumo. *Ferramentas de gestão de certificados eletrônicos são utilizadas por instituições de ensino e organizadores de eventos para simplificar e automatizar tarefas de emissão, entrega e validação de certificados para os participantes. A validação é tipicamente limitada a um token de autenticação estático, que é gerado de forma determinística (*e.g.*, resumo criptográfico) ou pseudo-aleatória (*e.g.*, conjunto de 8 letras quaisquer). Consequentemente, um atacante que conseguir comprometer o sistema pode comprometer (*e.g.*, adicionar novos ou modificar os existentes) os tokens de validação. Para mitigar este tipo de problema, propomos a ferramenta* `e-certsDS`*, que emite e publica certificados eletrônicos que podem ser validados através de uma assinatura digital OpenPGP e um código de autenticação adicional (utilizando a primitiva criptográfica HMAC). Os usuários finais podem verificar a autenticidade através da assinatura digital, o que impede o atacante de modificar certificados existentes (ou adicionar novos) sem ser detectado.*

1. Introdução

Atualmente existem diversos sistemas de geração de certificados eletrônicos para eventos, cursos, oficinas, entre outras atividades, como [da Silva Ribeiro et al., 2011, Sympla Internet Soluções S.A., 2021, Certifier, 2021, e-certificado, 2021, GC, 2021, KBR, 2021]. Alguns desses sistemas são gratuitos e de domínio público, como é o caso do SGCE [da Silva Ribeiro et al., 2011, DTIC, 2016], disponível no portal do software público brasileiro [Software Público Brasileiro, 2016].

Originalmente, o SGCE foi concebido para emissão de certificados dos participantes da ERRC 2010 [NTIC, 2010], e, apesar de ter sido desenvolvido em curto espaço de tempo, com objetivo de atender finalidades específicas do evento, vem sendo utilizado com sucesso pela Unipampa para gerenciar certificados eletrônicos de centenas de eventos e ações anuais da instituição. Desde que foi disponibilizado no portal do software público, o SGCE conta com mais de 5.700 downloads e está sendo utilizado por centenas de instituições no Brasil e em outros países.

Sistemas como o SGCE permitem a utilização de modelos de certificados e a geração automatizada de um grande número de certificados para listas de participantes dos eventos ou atividades. Tipicamente, como ilustrado no certificado da Figura 1, os certificados possuem uma URL (*e.g.*, `https://eventos.unipampa.edu.br/certificados/validar/78A9DA51`) de validação online (destacada em vermelho na Figura 1), que utiliza um token de autenticação (*i.e.*, 78A9DA51). Essa URL pode ser utilizada por terceiros para validar o documento. Entretanto, este mecanismo de verificação não oferece garantias relativas a integridade e a autenticidade dos arquivos

(PDF) dos certificados[1]. *O que acontece se o sistema (*e.g.*, SGCE), que concentra em um único local as informações dos certificados e os* tokens *de autenticação, for comprometido? Como recuperar/reconstruir os* tokens *de autenticação, que são frequentemente gerados por rotinas pseudo-aleatória?*

Figura 1. Exemplo de certificado eletrônico do SGCE

Em termos de segurança, é recomendado que sistemas utilizem assinaturas digitais ou códigos de autenticação baseados em primitivas criptográficas (*e.g.*, HMAC) que permitam verificação criptográfica da integridade e autenticidade do documento de forma pública (*e.g.*, assinatura digital) ou privada (*e.g.*, autenticação via HMAC, cuja chave secreta está em posse do emissor do documento). Por exemplo, as assinaturas digitais são verificáveis a partir de chaves públicas ou certificados digitais.

A ferramenta `e-certsDS` foi concebida como uma solução alternativa para a geração, publicação e validação de certificados eletrônicos, que são assinados utilizando uma assinatura digital no padrão OpenPGP (`https://www.openpgp.org`) e um código de verificação de autenticidade baseado na primitiva criptográfica HMAC. A assinatura digital OpenPGP permite a verificação pública da integridade e autenticidade dos certificados em PDF. O código de autenticação HMAC habilita um segundo nível de verificação privada, que pode ser utilizado em caso de comprometimento dos servidores de publicação dos certificados e respectivas assinaturas digitais.

O restante do paper está organizado como segue. Na Seção 2 apresentamos a organização e a implementação da ferramenta `e-certsDS`. Nas seções seguintes, apresentamos uma breve discussão (Seção 3), detalhes de planejamento da demonstração (Seção 4), algumas estatísticas de utilização da `e-certsDS` nos eventos e atividades de pesquisa, ensino e extensão do Programa Clube Universidade Hacker (`https://unihacker.club`) e trabalhos futuros (Seção 5).

[1]Neste paper utilizamos os termos “certificados eletrônicos” e “certificados” como sinônimos.

2. `e-certsDS`: organização e implementação

A Figura 2 ilustra a organização e a operação da `e-certsDS`. A ferramenta utiliza como entrada um template LaTeX (1), que contenha os componentes textuais e visuais, como logomarcas, informações sobre o evento e *tags* para preenchimento automático dos certificados. As *tags* são automaticamente substituídas pelos dados de entrada (2), representados por arquivos no formato CSV, que contém as informações básicas dos participantes do evento, como nome completo, endereço de email, tipo de participação e número de horas. A Figura 3 apresenta um exemplo de conteúdo de um arquivo CSV de entrada, onde cada linha contém os dados de um participante do evento. Este tipo de formato é bastante simples, e permite que os usuários utilizem ferramentas como o Google Forms para salvar os dados dos participantes em planilhas online, que posteriormente podem ser exportadas para arquivos CSV.

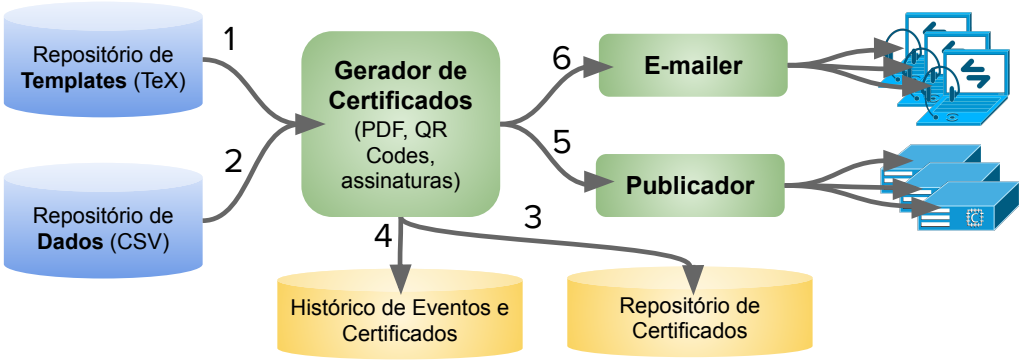

Figura 2. Organização e operação da `e-certsDS`

```
Alice Silva,alice@gmail.com,Co-Organizador,1
Bob Souza,bob@gmail.com,Ouvinte,2
Eve Martins,eve@gmail.com,Ouvinte,2
Charles Pinha,charles@gmail.com,Palestrante,2
Eder Soares,eder@gmail.com,Colaborador,1
João Bento,joao@gmail.com,Co-Organizador,1
Chico Costa,chico@gmail.com,Convidado Especial,2
```

Figura 3. Exemplo de conteúdo dos arquivos CSV

Quando executado, o gerador de certificados utiliza os arquivos de entrada e os parâmetros de execução, que incluem um identificador do evento (abreviatura do evento), a senha[2] para geração das assinaturas utilizando a chave privada OpenPGP, (c) a senha que é utilizada pela primitiva HMAC para gerar os códigos de autenticação, (d) a senha do banco de dados que contém o histórico de eventos e certificados gerados, e (e) modo de operação, que pode ser `teste` ou `deploy`. Caso o usuário queira simplificar a gestão das senhas, ele pode utilizar uma única senha para as três finalidades distintas. O banco de dados históricos foi implementado como sendo um arquivo de texto, que permanece compactado e protegido por senha. Quando necessário, o arquivo é decifrado, descompactado e utilizado para gravar novos registros históricos.

[2]Utilizaremos senha e chave secreta como sinônimos neste paper.

A seguir, é apresentado um exemplo de execução do gerador de certificados. A abreviatura do evento é utilizada para montar o assunto da mensagem que é enviada para cada participante do evento. O modo de operação indica se a ferramenta deve ou não realizar a publicação dos certificados (pelo *Publicador*) e enviar a notificação por email para os participantes (pelo *E-mailer*). No modo `teste`, a ferramenta gera os certificados para visualização e revisão. Após verificar se o conteúdo foi gerado adequadamente, o usuário pode executar novamente a ferramenta em modo `deploy`, quando os certificados são automaticamente publicados em uma lista (de 1 ou mais) de servidores *web* pré-configurada.

```
./gerador.sh templates/1.tex dados/1.csv AbrevEvento \
SenhaPGP SenhaHMAC SenhaHistorico teste
```

Na versão atual da ferramenta, os arquivos PDF dos certificados são publicados separadamente em servidores *web* utilizando a ferramenta `rsync` sobre um túnel SSH seguro, que utiliza chaves públicas para autenticação. Para cada um dos servidores envolvidos nesta publicação deve ser incluída uma configuração no arquivo padrão do SSH (`config`), na conta do usuário que opera a ferramenta. O arquivo `config` fica tipicamente em `.ssh/` no diretório do usuário. Na Figura 4 é apresentado um exemplo de configuração de dois servidores (`servidor1` e `servidor2`) remotos para publicação dos certificados. Utilizando o túnel SSH, o `rsync` necessita apenas sincronizar o diretório do repositório local de certificados com o diretório remoto de destino da publicação dos certificados nos servidores *web* (*e.g.*, `/var/www/certificados/` no `servidor1`).

```
Host servidor1
  hostname s1.certificados.org
  user publicador
  port 22
  identitiesonly yes
  PubkeyAuthentication yes
  identityfile ~/.ssh/id_rsa_s1

Host servidor2
  hostname s2.certificados.org
  user publicador
  port 22
  identitiesonly yes
  PubkeyAuthentication yes
  identityfile ~/.ssh/id_rsa_s2
```

Figura 4. Exemplo de configuração do `config` do SSH

No email enviado pela ferramenta, o participante do evento receberá o link do PDF do certificado (*e.g.*, `https://certificado.unihacker.club/20200611/97658b389b9be5ed568f95cb98a6ad0e.pdf`. Na versão atual da ferramenta, um certificado `e-certsDS` possui cinco QR Codes, como ilustrado no exemplo da Figura 5. Cada QR Code contém uma informação, conforme detalhado a seguir.

1. link do PDF original do certificado;
2. link do arquivo contendo o resumo criptográfico SHA256 do PDF;
3. link do arquivo ".asc", que contém a assinatura digital OpenPGP do certificado;
4. identificador da chave pública OpenPGP e instruções para *download*;
5. resumo criptográfico do código de autenticação HMAC do certificado.

Os três primeiros QR Codes seguem o padrão de distribuição dos arquivos de instalação (arquivos ISO) de distribuições GNU/Linux. Cada arquivo ".iso" acompanha um arquivo de resumos criptográficos e um segundo arquivo ".asc" para verificação da assinatura OpenPGP. Com estas informações em mãos, assumindo que o responsável

Figura 5. Exemplo de certificado eletrônico da `e-certsDS`

pela assinatura digital dos certificados é pré-definido e conhecido (*i.e.*, nome e email), os usuários finais podem verificar a integridade e a autenticidade dos certificados.

O quarto QR Code é meramente técnico e opcional, uma vez que a chave pública OpenPGP pode ser recuperada através do email do autor da assinatura digital. Finalmente, o último QR Code apresenta o resumo criptográfico resultante da aplicação da primitiva HMAC, utilizando uma chave secreta conhecida apenas pelo emissor dos certificados, sobre os dados dos outros quatro QR Codes. Em outras palavras, o quinto QR Code autentica o conteúdo dos demais.

O `e-certsDS` foi implementado utilizando as linguagens Bash Scripting e Python (pacotes/módulos: `smtplib`, `email`, `sys`, `random`, `time`). Para o enviador de emails, implementado em Python, é importante que haja um tempo pseudo-aleatório de espera (*e.g.*, de 1s a 12s) entre o envio de um email e outro. Isto reduz a probabilidade de bloqueio automático simples (*e.g.*, baseado em frequência de mensagens) frequentemente implementado por provedores de email. A maior parte da ferramenta foi implementa utilizando Bash Scripting e diversas ferramentas e comandos disponíveis em sistemas GNU/Linux, como `7z`, `qrencode`, `shasum`, `gpg2`, `rsync`, e `pdflatex`. A versão 0.2 (beta) da ferramenta está disponível em `https://github.com/uhc-ec/e-certsDS`.

3. Discussões e Desafios

Neste seção apresentamos algumas discussões e desafios relacionados com questões de armazenamento, usabilidade e segurança dos certificados eletrônicos. Acreditamos que essas discussões podem contribuir com a evolução da `e-certsDS` e o desenvolvimento de sistemas similares.

Armazenamento

Diferentemente de outras soluções, como o SGCE, que armazenam apenas informações essenciais e templates dos certificados, o `e-certsDS` armazena localmente (*i.e.*, na máquina do responsável pela emissão dos certificados) os dados de entrada e o histórico dos certificados gerados. Os arquivos PDF dos certificados são publicados separadamente, em servidores *web*. Considerando o histórico de 840 certificados já gerados para o Programa Clube Universidade Hacker, que ocupam 135MB de espaço em disco, cada certificado ocupa em média 161KB. Se considerarmos que o armazenamento é um dos recursos computacionais mais baratos e abundantes hoje em dia, acreditamos ser uma solução viável para cenários de pequena e média escala.

Alternativamente, caso o volume de certificados seja grande, isto é, o armazenamento seja um desafio, os certificados podem ser enviados por email para os participantes dos eventos ou atividades. Ao invés de armazenar os certificados em servidores *web*, o `e-certsDS` poderia armazenar apenas a assinatura digital (além dos dados de entrada, que são armazenados separadamente).

Usabilidade

Atualmente, a verificação dos dados contidos nos QR Codes envolve conhecimentos técnicos básicos, como validar assinaturas utilizando ferramentas OpenPGP. Para melhorar a usabilidade da ferramenta, sob a perspectiva do usuário final, é interessante que exista um aplicativo específico para validação da integridade e autenticidade dos certificados. O aplicativo, de forma simples, deve conter a chave pública OpenPGP para realizar a rápida verificação da autenticidade da assinatura do PDF do certificado. Além da simplicidade e rapidez, o aplicativo permite aumentar a confiabilidade do processo, tornando desnecessário entrar em contato com o responsável (*e.g.*, nome e email) pela assinatura OpenPGP.

Embutir a assinatura digital no próprio arquivo PDF do certificado, assim como ocorre nos aplicativos de leitura de arquivos PDF que permitem inserir e verificar assinaturas digitais, pode ser outro aspecto para melhorar a usabilidade da solução. Neste caso, a assinatura digital deve ser realizada sobre partes do conteúdo do PDF e não sobre todo o arquivo, como é realizado atualmente. Essa assinatura embutida e o aplicativo de verificação comentado anteriormente resultariam, na nossa opinião, em uma versão mais simples, mais usável e mais robusta da solução.

Segurança

A `e-certsDS` melhora aspectos de segurança na gestão de certificados eletrônicos por pelo menos duas razões. Primeiramente, a solução utiliza assinaturas digitais publicamente verificáveis aos invés de *tokens* de autenticação gerados pseudo-aleatoriamente. A segurança desses *tokens* depende tipicamente da robustez e da segurança de um ponto

único de falhas, o servidor do sistema. Em segundo lugar, o processo de emissão e publicação dos certificados é executado localmente, sem interface publicamente acessível (*e.g.*, interface *web*). Os certificados são publicados em servidores *web* remotos. Estes dois pontos, quando comparados com sistemas tradicionais, como o SGCE, reduzem substancialmente a superfície de ataque e diminuem consideravelmente a probabilidade de vulnerabilidades remotamente exploráveis.

Soluções como o SGCE funcionam de forma online e possuem uma base de código e bibliotecas externas que podem comprometer a segurança do sistema. Por exemplo, o SGCE necessita a versão 5 do PHP e diversas bibliotecas que não são mais sequer suportadas pela comunidade, o que aumenta ainda mais os riscos de incidentes de segurança de difícil resolução. Eventualmente, em caso de uma vulnerabilidade severa, de fácil exploração, poderia ser necessário a migração do sistema para a versão 8 do PHP.

No caso da solução `e-certsDS`, comprometer os servidores dos certificados representa um desafio maior, pois utilizamos apenas os recursos mínimos de servidores *web*, o que favorece a atualização constante (*e.g.*, novas versões ou diferentes servidores *web*), a segurança e a robustez do sistema. É evidente que o risco de segurança, ou comprometimento da solução, mudou de local, isto é, ao invés de ser o servidor *web* (ou aplicação *web*), passou a ser a máquina utilizada pelo emissor dos certificados.

4. Demonstração

A demonstração da ferramenta será realizada através de um ambiente virtual composto por três máquinas virtuais, uma para instanciar e utilizar o `e-certsDS` e outras duas atuando como servidores *web* de armazenamento dos certificados eletrônicos. O ambiente estará hospedado em dispositivo próprio dos autores. O funcionamento da ferramenta será demonstrado através dos seguintes passos:

1. apresentação dos requisitos de instalação e configuração;
2. apresentação da configuração dos ambientes das máquinas virtuais;
3. apresentação dos arquivos de entrada (*template* e arquivo CSV com as informações dos participantes);
4. demonstração do processo de emissão e publicação dos certificados eletrônicos;
5. demonstração de acesso e verificação de integridade e autenticidade dos certificados eletrônicos através da assinatura digital OpenPGP;
6. demonstração da verificação do código de autenticação HMAC, que pode ser utilizado para validar a integridade e autenticidade dos demais QR Codes do certificado.

5. Considerações Finais

Conclusão

A ferramenta `e-certsDS` (`https://github.com/uhc-ec/e-certsDS`), em sua versão 0.2 (beta), vem sendo utilizado pelo Programa Clube Universidade Hacker (`https://UniHacker.Club`) para a emissão e publicação de certificados eletrônicos. Atualmente, há 840 certificados, para mais de 40 atividades distintas, incluindo eventos, oficinas, palestras, treinamentos e cursos, registrados e publicados no sistema.

A e-certsDS foi concebida para dar um nível mais robusto de segurança à emissão de certificados eletrônicos através da inclusão de assinaturas digitais e códigos de autenticação HMAC. Com a utilização de assinaturas digitais OpenPGP, a ferramenta permite a verificação descentralizada e mais confiável da integridade e autenticidade dos certificados. Em posse do PDF do certificado e do arquivo da assinatura digital, qualquer usuário pode verificar a autenticidade do certificado sem depender da disponibilidade e da confiabilidade necessários às soluções tradicionais, como SGCE, Certifier e Sympla. Apesar de estar em uma versão experimental, plenamente funcional, acreditamos que o e-certsDS pode evoluir em termos de usabilidade, bem como incorporar novas características e recursos de segurança e privacidade. Alguns exemplos são apresentados na sequência, na lista de trabalhos futuros.

Trabalhos Futuros

Como trabalhos futuros podemos elencar:

1. análises de segurança da ferramenta sob diferentes cenários e condições de ataque;
2. aplicativo móvel para simplificar e agilizar a verificação da integridade e autenticidade dos certificados eletrônicos (*e.g.*, através da leitura do primeiro QR Code ou do próprio arquivo PDF);
3. interface visual para simplificar a utilização da ferramenta para usuários leigos;
4. embutir a assinatura digital OpenPGP no próprio arquivo PDF do certificado (isto implica em algumas mudanças no processo de geração e validação da assinatura digital, que não poderá mais ser do PDF por completo, como é hoje);
5. adicionar a chave pública OpenPGP ao arquivo PDF do certificado, eliminando a necessidade de realizar o download da chave de um servidor público PGP;
6. permitir a utilização de certificados ICPEdu para assinatura digital dos certificados eletrônicos gerados pela ferramenta.

Referências

Certifier (2021). Online certificate creator. https://certifier.me.

da Silva Ribeiro, P., Conrad, P., Junior, S. A. M. B., and Kreutz, D. (2011). Sistema de Gestão de Certificados Eletrônicos. In *V Workshop de TIC das IFES*, Florianópolis.

DTIC (2016). Sistemas de Gestão de Certificados Eletrônicos (SGCE). Universidade Federal do Pampa (Unipampa). Manual do Usuário.

e-certificado (2021). Certificados online para cursos. https://e-certificado.com.

GC (2021). Gerador de certificados. https://geradordecertificados.com.

KBR (2021). Online bulk certificate maker software. https://www.edusys.co/en-in/college-certificate-making-software.html.

NTIC (2010). ERRC 2010 encerra com palestras e premiação. https://dtic.unipampa.edu.br/2010/10/13/.

Software Público Brasileiro (2016). SGCE - Sistema de Gestão de Certificados Eletrônicos. https://softwarepublico.gov.br/social/sgce.

Sympla Internet Soluções S.A. (2021). Sympla. https://www.sympla.com.br.