

Mestrado em Engenharia Informática (MEI)

Mestrado Integrado em Engenharia Informática

(MiEI)

Perfil de Especialização **CSI** : Criptografia e Segurança da Informação

Engenharia de Segurança

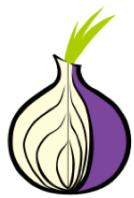


Tópicos

- Criptografia Aplicada
 - Protocolos/aplicações criptográficas
 - TOR (The Onion Router)
 - Voto eletrónico

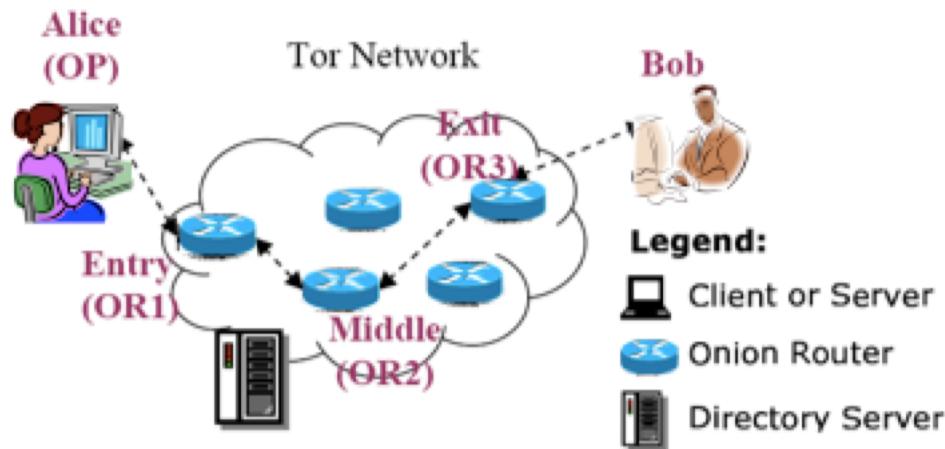
Motivação

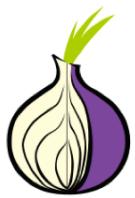
- Alguns destes *items* de criptografia aplicada poderão ser necessários para o projeto de desenvolvimento de software (“Leilões online” e “Gestor de passwords com base em QrCodes”).



TOR (The Onion Router)

- **Protocolo criptográfico de rede** cujo objectivo é:
 - Garantir a **anonimidade ponto-a-ponto** (ao nível não aplicacional) de um utilizador na Internet;
 - Note que todos os protocolos que vimos até agora estabelecem túneis seguros/privados/confidenciais, mas não permitem anonimidade ponto-a-ponto (ao nível não aplicacional), na medida em que os *headers* dos pacotes (TCP / UDP / IP) ainda revelam muita informação sobre o utilizador.
 - Permitir disponibilizar **serviços anónimos** (*hidden services*) – www e outros serviços – sem revelar a localização dos mesmos.





TOR (The Onion Router)

- Rede sobreposta à Internet constituída por ***Onion Routers* (OR)**
 - Cada OR executa como um processo normal do utilizador sem necessidade de privilégios especiais;
 - Cada OR conecta-se a outros OR através de uma conexão TLS;
 - Existem OR “mais confiáveis” que actuam como *Directory Server* – fornecem listas assinadas dos OR conhecidos e seu estado actual, que são descarregadas periodicamente pelos utilizadores do TOR –;
 - Cada OR tem um par de chaves de identidade de longo tempo (*identity key*) utilizado para assinar os certificados TLS, o descriptor do OR (contém chaves públicas, endereço, largura de banda, política de saída, etc.) e a directoria (no caso dos *Directory Server*);
 - Cada OR tem um par de chaves de curto prazo (*onion key*), rodadas periodicamente, utilizado para estabelecer chaves de sessão com o utilizador (através de Diffie-Hellman).
- Cada utilizador executa um software local: ***Onion Proxy* (OP)**
 - OP obtém dados da directoria, estabelece circuitos através da rede TOR e gere conexões das aplicações do utilizador.

How Tor Works:

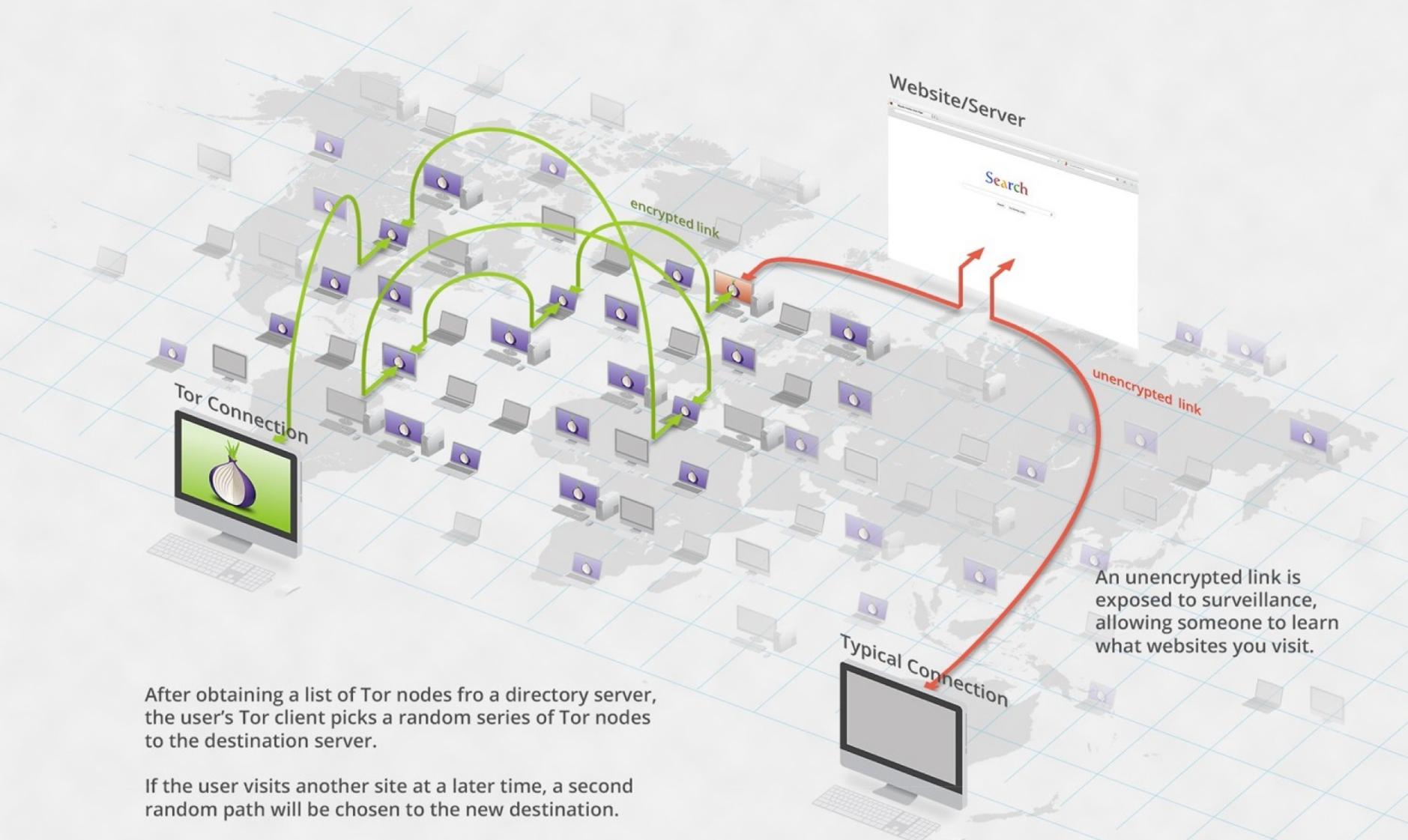
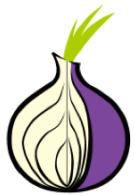


Imagen: <https://www.extremetech.com/internet/226106-onionscan-tests-dark-web-sites-to-see-if-they-really-are-anonymous>



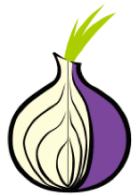


TOR (The Onion Router)

- OR comunicam entre si e com OP através de conexões TLS em pacotes (células) de tamanho fixo.
 - Cada célula tem 512 bytes e é constituída por um *header* e um *payload*;
 - O *header* inclui identificador do circuito circID que indica a que circuito a célula se refere (vários circuitos podem ser multiplexados sobre a mesma ligação TLS) e um comando CMD que descreve o que fazer com o *payload*;



- De acordo com o CMD, as células podem ser *control cells* (sempre interpretadas pelo OR que as recebe) ou *relay cells* (levam dados ponto a ponto);
- O CMD das *control cells* pode ser:
 - padding (utilizado para *keepalive*),
 - create/created (para criar novo circuito), ou
 - destroy (para finalizar circuito);

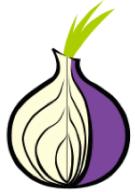


TOR (The Onion Router)

- OR comunicam entre si e com OP através de conexões TLS em pacotes (células) de tamanho fixo.
 - As *relay cells* têm um *header* adicional com o streamID (identificador do stream: vários streams podem ser multiplexados sobre o mesmo circuito), hash ponto-a-ponto (para verificação de integridade), o tamanho do payload do relay e o CMD do relay;

2	1	2	6	2	1	498
CircID	Relay	StreamID	Digest	Len	CMD	DATA
 - Todo o conteúdo do relay header e relay payload (i.e., o payload da célula) é cifrado ou decifrado (AES 128 bits) sequencialmente à medida que a célula se move ao longo do circuito;
 - O CMD do relay pode ser:
 - data (para dados a serem comunicados na stream),
 - begin (para abrir nova stream), end (para fechar stream),
 - teardown (para fechar uma stream “estragada”),
 - connected (para notificar o OP que foi efectuado um begin com sucesso),
 - extend/extended (para extender o circuito por mais um OR),
 - truncate/truncated (para destruir apenas parte do circuito),
 - sendme (para controlo de congestionamento num OR), ou
 - drop (para testar stream);





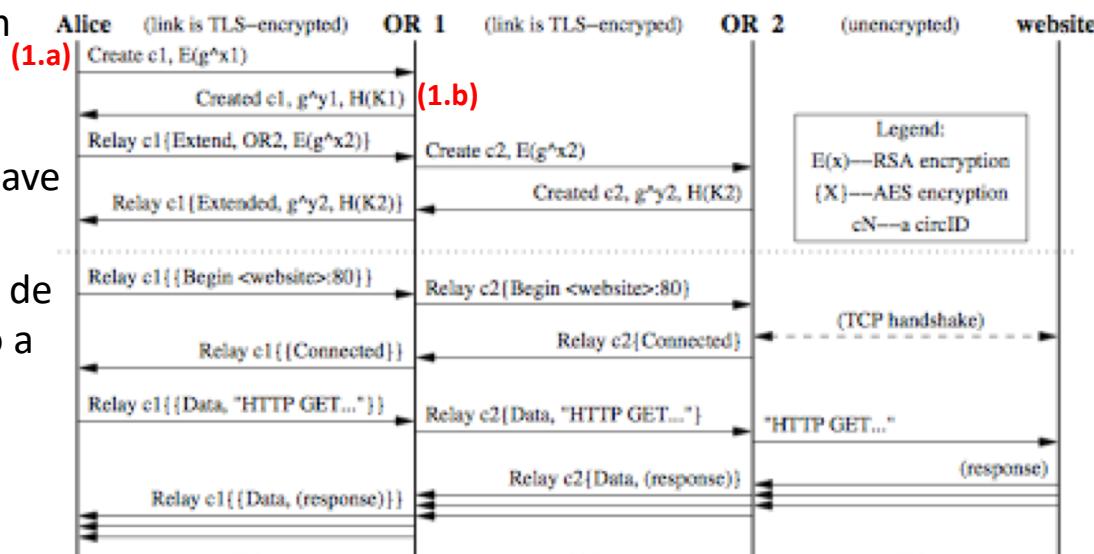
TOR - Anonimização

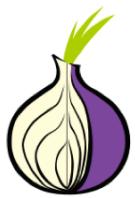
- OP pré-estabelece circuitos (normalmente de 3 OR) e muda para um novo circuito uma vez por minuto, garantindo que apenas um número limitado de pedidos podem ser ligados uns aos outros no OR de saída;
- OP constrói o circuito incrementalmente, negociando uma chave simétrica com cada OR do circuito, OR a OR. Note-se que escolhe os OR a partir da lista de OR fornecida pelo *Directory Server* que tem associado as chaves de longo termo (*identity key* para assinar certificados TLS) e de curto termo (*onion key* para estabelecer chaves de sessão por Diffie-Hellman).

Passo 1a: O OP (Alice) envia uma célula de controlo *create* para o primeiro nodo (OR1) do caminho escolhido pelo OP, escolhendo um novo cirID e com o payload da célula contendo a primeira metade da troca de chaves Diffie-Hellman (g^{x1} cifrado com a chave pública da *onion key* de OR1).

Passo 1b: O OR1 responde com uma célula de controlo *created*, contendo g^{y1} assim como a hash da chave K_1 negociada ($K_1 = g^{x1,y1}$).

A partir deste momento, OP e OR1 podem comunicar a célula de *relay* com o payload cifrado com a chave K_1 .





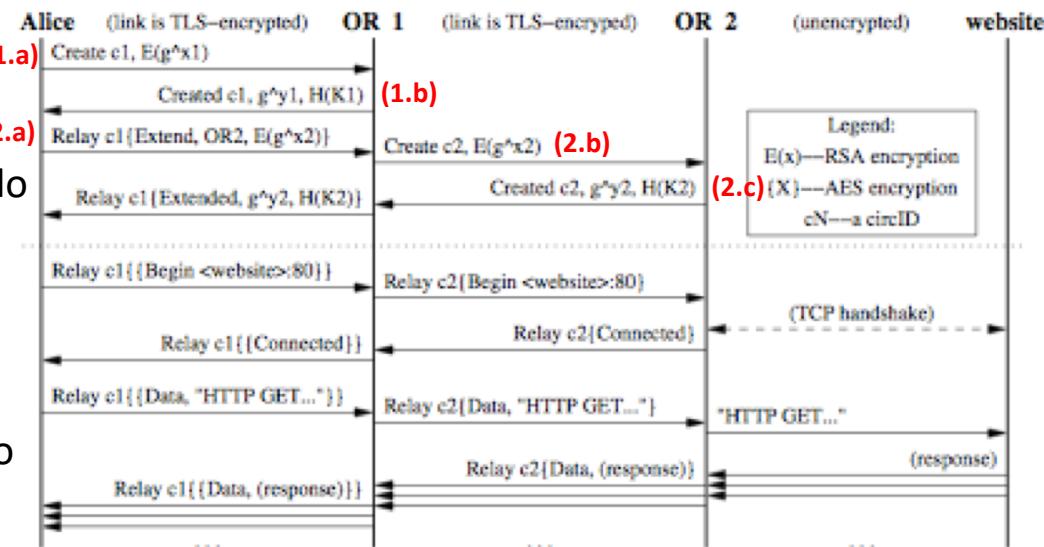
TOR - Anonimização

Passo 2a: Para extender o circuito, o OP (Alice) envia uma célula de *relay extend* ao OR1, identificando o próximo OR (OR2) e com g^{x^2} cifrado com a chave pública da *onion key* de OR2 ($E(g^{x^2})$).

Passo 2b: OR1 escolhe um novo CircID, copia (2.a) $E(g^{x^2})$ para o payload de uma célula de controlo *create* e, envia-a a OR2.

Passo 2c: OR2 responde com uma célula de controlo *created* e OR1 copia o payload dessa célula para uma célula de *relay extended* que envia a OP. O circuito está extendido a OR2 e o OP e OR2 partilham a chave comum $K_2 = g^{x^2 \cdot y^2}$.

A partir deste momento, OP e OR2 podem comunicar a célula de *relay* com o payload cifrado com a chave K_2 .



Para extender o circuito para nodos adicionais (OR_{n+1}), OP (Alice) efetua os passos anteriores, indicando sempre ao último OR (OR_n) no circuito para extender ao novo OR (OR_{n+1}).

TOR - Anonimização

O Protocolo de estabelecimento do circuito garante autenticação unilateral (OP sabe que está a trocar chaves com o OR, mas o OR não sabe quem está a abrir o circuito – i.e., OP não usa a sua chave pública e mantém-se anónimo).

Assim que o circuito está estabelecido, OP pode enviar células de *relay* até ao último OR do circuito.

(1.a)
(2.a)
(3.a)

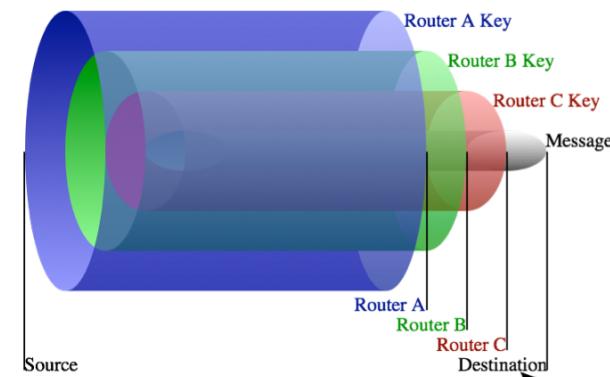
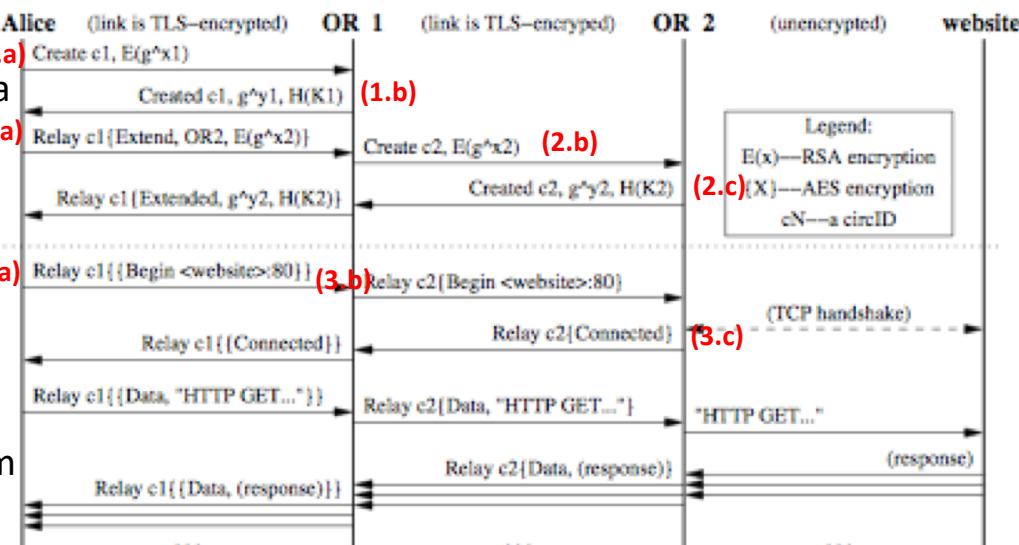
(1.b)
(2.b)
(3.b)

(2.c)
(3.c)

Passo 3a: OP (Alice) envia a célula de *relay* ao OR destinatário. Para construir essa célula, insere-lhe todos os dados necessários, gera o hash/digest e cifra com cada chave simétrica trocada com cada um dos OR (neste caso cifra com OR1 e depois com OR2).

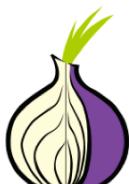
Passo 3b: o primeiro OR (OR1) decifra o payload e verifica se o hash está correcto. Se estiver, efectua o comando pedido pelo OP. Se não estiver, pega no payload decifrado e envia uma célula de *relay* para o OR seguinte no circuito.

Passo 3c: O OR final responde, através do circuito estabelecido, com uma célula de *relay* com o payload cifrado para o OP. Os OR intermédios adicionam novos níveis de cifra à medida que a “reencaminham” para o OP.



TOR – Pontos de *Rendezvous* e serviços anónimos

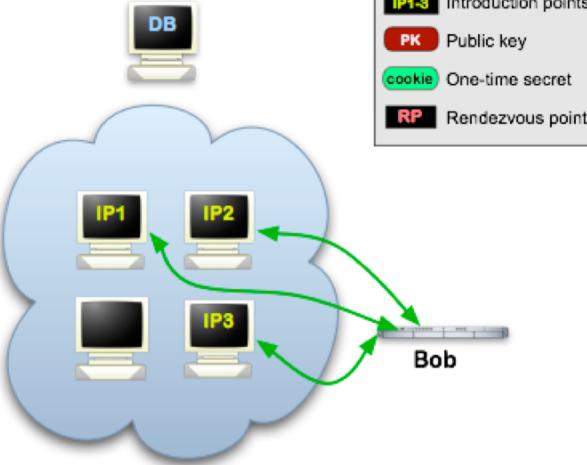
- Pontos de *Rendezvous* são o suporte para a disponibilização de serviços anónimos (também designados por *resolver anonymity*).
 - Na rede TOR, a disponibilização de serviços anónimos permite a um OP (Bob) disponibilizar serviços TCP (por exemplo, servidor web) sem revelar o seu endereço IP.
 - Como o OP (Alice) que acede ao serviço anónimo também é anonimizado, tanto o OP que acede como o OP que é acedido são anónimos.



TOR – Pontos de Rendezvous e serviços anónimos

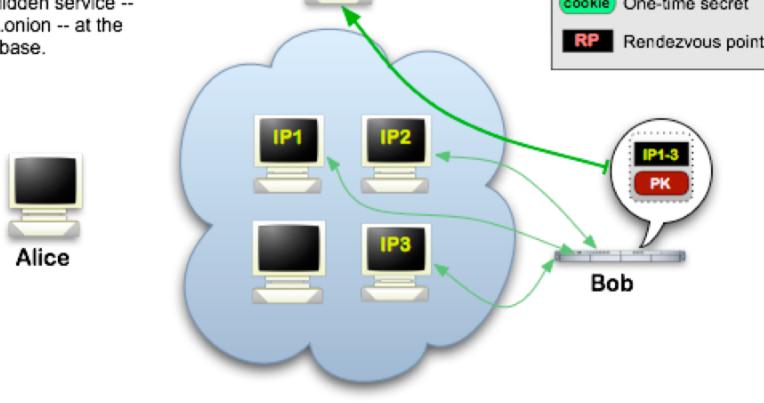
Tor Hidden Services: 1

Step 1: Bob picks some introduction points and builds circuits to them.



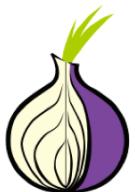
Tor Hidden Services: 2

Step 2: Bob advertises his hidden service -- XYZ.onion -- at the database.



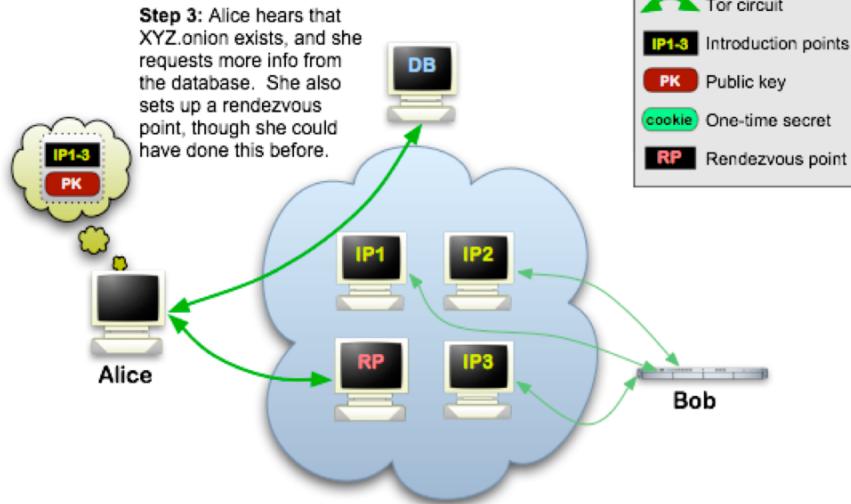
Passos 1 e 2:

- Bob gera um par de chaves de longo tempo para identificar o seu serviço Web (a chave pública passa a ser o identificador do serviço);
- Bob escolhe alguns pontos de introdução (*introduction points*) e anuncia-os no *Directory Server*, assinando o anúncio (descriptor do serviço) com a sua chave privada;
 - O descriptor do serviço anônimo contém a chave pública do serviço e um sumário de cada ponto de introdução;
 - O descriptor/serviço será encontrado pelos clientes que acederem a XYZ.onion na rede TOR, onde XYZ é um nome de 16 caracteres derivado da chave pública do serviço.
- Bob cria um circuito TOR (conforme visto nos últimos slides) para cada um dos pontos de introdução e pede-lhes para esperarem por pedidos.

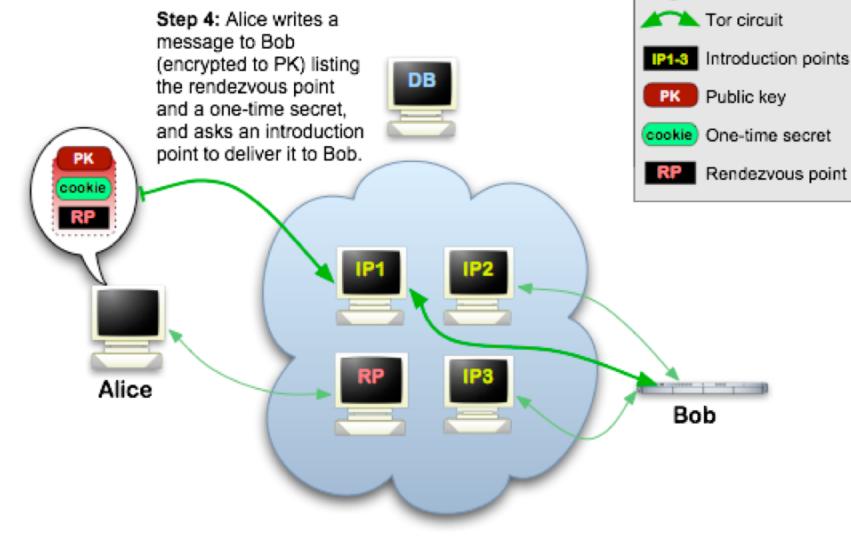


TOR – Pontos de Rendezvous e serviços anónimos

Tor Hidden Services: 3



Tor Hidden Services: 4



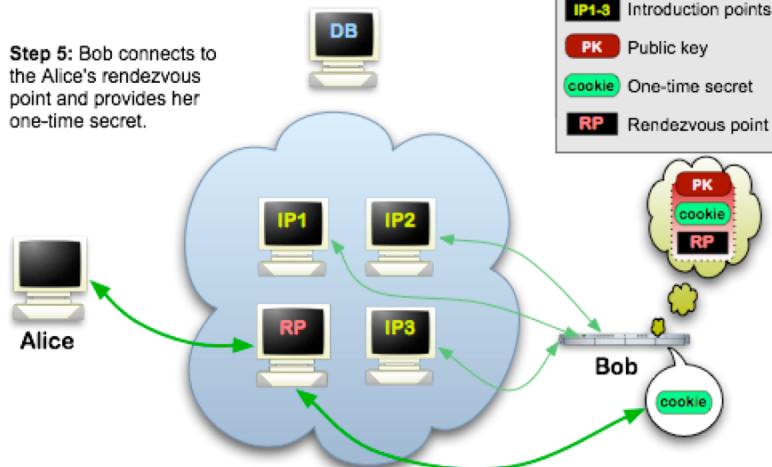
Passos 3 e 4:

- Alice sabe da existência do serviço XYZ.onion e acede aos seus detalhes (chave pública e pontos de introdução) no TOR através do Directory Server;
- Alice escolhe um OR como ponto de *rendezvous* (RP) para a conexão com o serviço XYZ.onion;
- Alice constrói um circuito TOR até ao RP e fornece-lhe um “*rendezvous cookie*” (segredo aleatório único) para posterior reconhecimento do serviço XYZ.onion;
- Alice abre um *stream* anónimo até um dos pontos de introdução e fornece-lhe uma mensagem (cifrada com a chave pública de XYZ.onion) com informação sobre o RP, o “*rendezvous cookie*” e o inicio de troca de chaves Diffie-Hellman. O ponto de introdução reencaminha a mensagem para o serviço XYZ.onion através do circuito TOR criado nos passos anteriores.

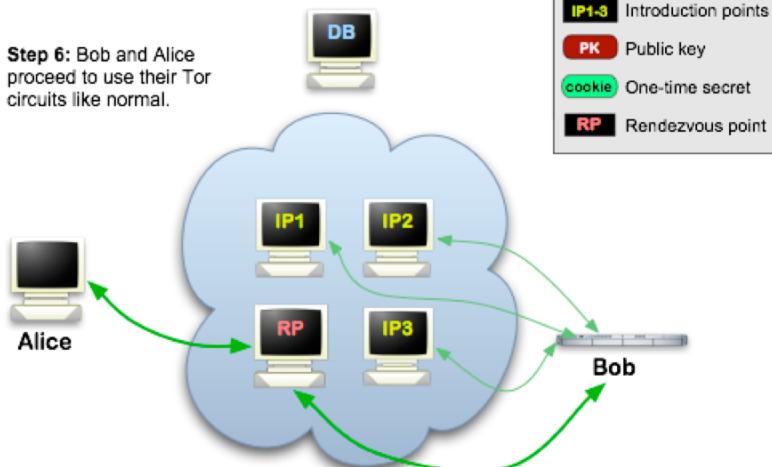


TOR – Pontos de Rendezvous e serviços anónimos

Tor Hidden Services: 5



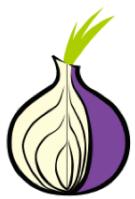
Tor Hidden Services: 6



Passos 5 e 6:

- Se Bob (XYZ.onion) pretender falar com Alice (OP), Bob cria um circuito TOR até ao RP e envia o “rendezvous cookie”, a segunda parte da troca de chaves Diffie-Hellman e um *hash* da chave de sessão que agora partilha com Alice;
- O RP conecta o circuito de Alice com o circuito de Bob (normalmente o circuito consiste de 6 OR: 3 escolhidos por Alice sendo o terceiro o RP e, outros 3 escolhidos por Bob). Note-se que o RP não consegue reconhecer Alice, Bob nem os dados que transmitem;
- Alice envia uma célula de *relay begin* através do circuito que, ao chegar ao OP de Bob, conecta com o servidor Web de Bob;
- Um *stream* anónimo foi estabelecido e Alice e Bob comunicam da forma normal num *stream* TOR.





TOR – Possíveis Ataques

O que acontece se o nosso servidor actuar como nodo de entrada?

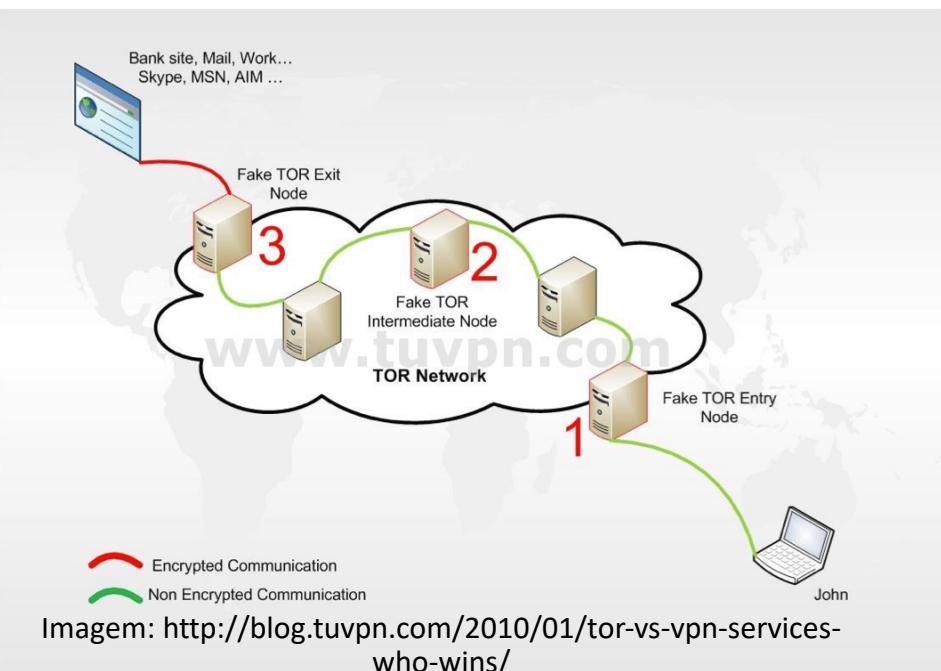
- Sabe que o IP de John está a utilizar a rede TOR (interessante para SIGINT – *signal intelligence* – mas pouco mais).

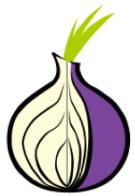
O que acontece se o nosso servidor actuar como nodo intermédio?

- Nada. Recebe informação cifrada de um nodo TOR que vai para outro nodo TOR.

O que acontece se o nosso servidor actuar como nodo de saída?

- A missão do nodo de saída é decifrar a comunicação, e enviá-la para o serviço (web) de destino, pelo que não sabe quem foi o originante da comunicação mas sabe qual é a comunicação.
- Podemos argumentar que se o destino é SSL, não há hipótese de aceder à comunicação em claro do John.
- Errado. Foi demonstrado que é possível um *MitM attack* sobre o SSL, no nodo de saída.





TOR – Possíveis Ataques

O que acontece se um servidor nosso actuar como nodo de entrada e outro servidor nosso actuar como nodo de saída?

- É possível calcular coeficientes de correlação – através de fórmulas matemáticas da área da probabilidade e estatística – na análise de pacotes nos dois nodos, baseado na frequência, timing e tamanho do pacote.
- Supõe-se que 80% dos utilizadores podem ser de-anonimizados no período de 6 meses.

- Custo elevado, já que uma grande percentagem dos OR (TOR tem cerca de 7.000 OR) têm que pertencer à entidade que queira efetuar um ataque de correlação.
- Utilizado por agências governamentais (BND, GCHQ, NSA).
- Supõe-se que o “Silk Road 2.0” foi de-anonimizado com base neste ataque.

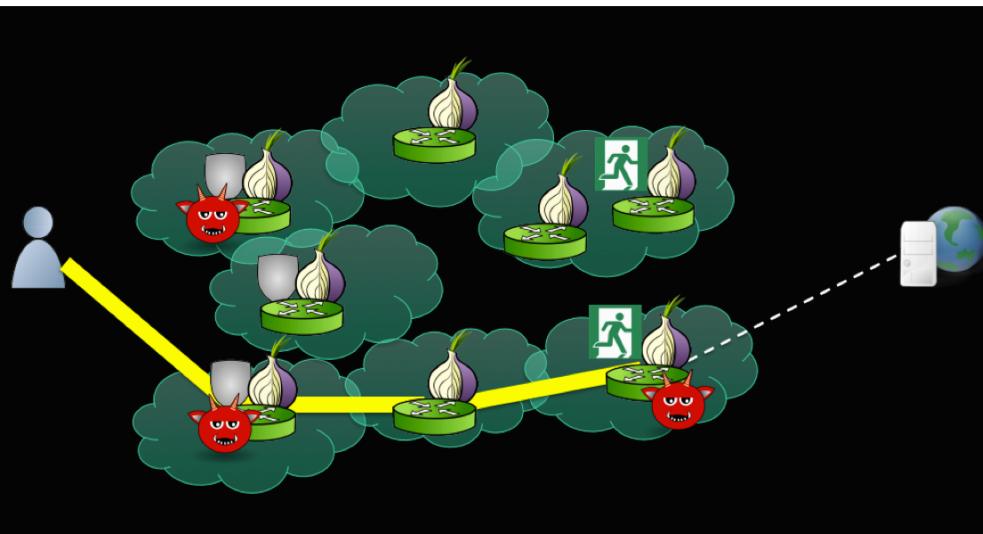


Imagen: <https://www.deepdotweb.com/2016/10/25/tors-biggest-threat-correlation-attack/>

TOR

- Exemplo efectuado em menos de dez minutos, a partir da mesma máquina, no site www.whatismyip.com

Tor Browser

IP Address: 87.106.148.90
City: Karlsruhe
State/Region: Baden-wurttemberg
Country Code: DE
Postal Code: 76229
ISP: 1&1 Internet Ag
Time Zone: +01:00 UTC/GMT
Latitude: 49.0047
Longitude: 8.3858

IP Address: 92.222.172.41
City: Roubaix
State/Region: Nord-pas-de-calais
Country Code: FR
Postal Code: 59689
ISP: Ovh Sas
Time Zone: +01:00 UTC/GMT
Latitude: 50.6942
Longitude: 3.1746

Firefox

IP Address: 94.242.206.170
City: Steinsel
State/Region: Luxembourg
Country Code: LU
Postal Code: I-7349
ISP: Root Sa
Time Zone: +01:00 UTC/GMT
Latitude: 49.6769
Longitude: 6.1239

IP Address: 94.132.113.217
City: Porto
State/Region: Porto
Country Code: PT
Postal Code: -
ISP: TVCABO Portugal, S.A.
Latitude: 41.1496
Longitude: -8.611

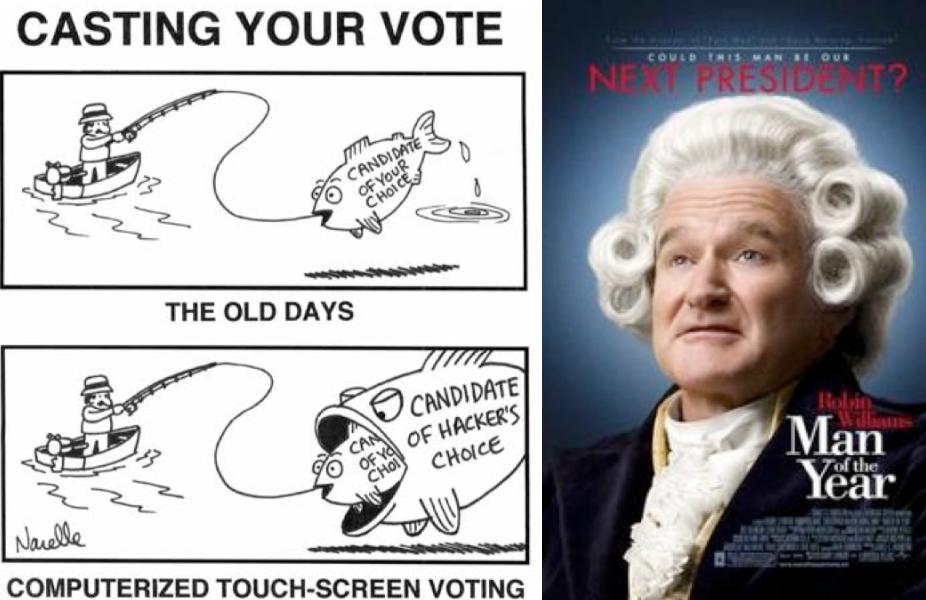
- Qual a explicação?



Voto Electrónico

- Protocolo de votação em rede aberta, com as seguintes **garantias gerais**:

- Autenticação do Eleitor
- Anonimato do Eleitor
- Confidencialidade do Voto
- Integridade do Voto
- Não extravio do Voto
- Integridade do sistema de Voto
- Auditabilidade do sistema de Voto



- O protocolo dos exemplos apresentados assume que os votantes não dispõem de meios de identificação avançados (como por exemplo, certificado digital pessoal) nem são peritos tecnológicos.

Voto Electrónico – Exemplo 1

- **Entidades** participantes no Voto Electrónico:
 - Promotor: O indivíduo/entidade que promove a votação;
 - SMV (Sistema de Mediação de Votos): plataforma de votação (*frontend* e *backend*) responsável por gerir e processar os votos electrónicos durante o período de votação, findo o qual os fará chegar ao Promotor;
 - Votante: O indivíduo que pretende exercer o seu direito de voto, escolhendo as opções existentes, conforme política do boletim de voto (para efeitos do protocolo, voto branco e voto nulo podem ser vistos como opções adicionais que o Promotor pode ou não activar);
 - Auditor: Indivíduo/Entidade que acompanha o processo de votação, podendo comprovar a correcção deste sem tomar conhecimento sobre o sentido de voto de cada um dos participantes.

Voto Electrónico – Exemplo 1

- Com este protocolo de Voto electrónico pretende-se:
 - O Votante possa aceder a interface Web ou app de votação que, após lhe mostrar as opções existentes e solicitar a introdução dos elementos de autenticação necessários, lhe permita votar e obter uma prova de como efectuou a votação dentro do prazo estipulado;
 - O Votante possa provar a existência de situações de rejeição fraudulenta de votos efectuados dentro do prazo estabelecido;
 - O Votante possa proteger-se do facto de, embora tenha efectuado o seu voto dentro do período da votação, o servidor apenas o tenha processado após este período ter terminado;
 - O SMV não tome conhecimento do conteúdo dos votos;
 - O Promotor não tome conhecimento do conteúdo dos votos até ao final da eleição;
 - Nenhum interveniente no processo consiga saber quem votou o quê;
 - O SMV não possa alterar o conteúdo dos votos sem que isso seja detectado pelo Promotor;
 - O SMV não possa alterar o número de votos sem que isso seja detectado pelo Promotor;
 - O Promotor não possa alterar o conteúdo dos votos sem que isso seja detectado pelo SMV;
 - O Promotor não possa alterar o número de votos sem que isso seja detectado pelo SMV;
 - O Promotor receba, de forma segura, os votos (cifrados com a chave pública do Promotor) efectuados durante o período de votação, juntamente com um comprovativo do instante temporal em que deram entrada no SMV;
 - O Promotor se possa defender de suspeitas infundadas de rejeição de votos efectuados dentro do prazo estabelecido;
 - O Promotor possa ter a certeza que durante o período da votação ninguém toma conhecimento do conteúdo dos votos.



Voto Electrónico – Exemplo 1

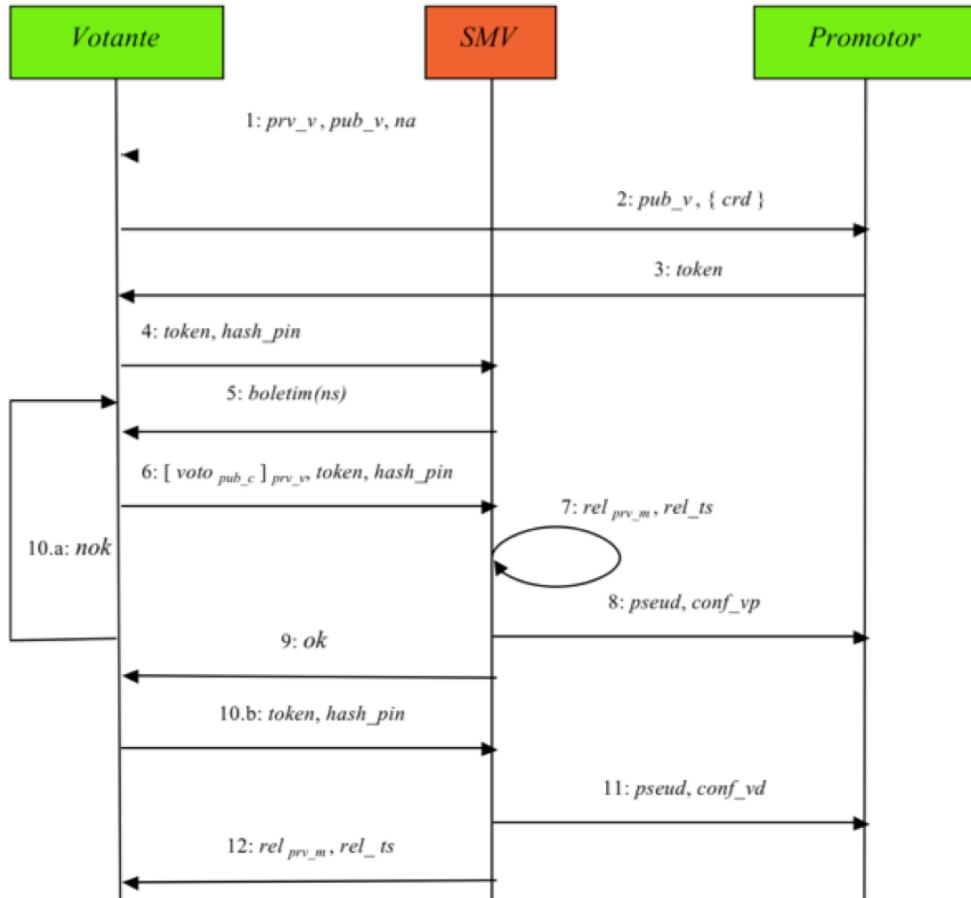
- Preparação do Voto electrónico (fase pré-voto):
 - O Promotor comunica ao SMV: grupo de votantes, período de votação, boletins de voto e possibilidades de voto de cada boletim (política do boletim de voto).
 - O Promotor comunica ao SMV: a identificação dos votantes e contactos, de modo ao SMV enviar credenciais de autenticação de votação (por exemplo, número de identificação do votante perante o Promotor e PIN gerada pelo SMV);
 - O Promotor indica ao SMV que tipo de credencial adicional de autenticação deve ser pedido ao Votante de modo ao Promotor a conseguir validar (por exemplo, número de identificação do votante perante o Promotor e data de nascimento);
 - O Promotor gera um par de chaves e o respectivo certificado de votação, fornecendo o certificado ao SMV;
 - O Promotor gera um par de chaves e o respectivo certificado de assinatura, fornecendo o certificado ao SMV;
 - O Promotor gera um par de chaves e o respectivo certificado de servidor Web, fornecendo o certificado ao SMV (para cifra dos votos);
 - O Promotor disponibiliza um serviço Web HTTPS que, recebendo uma chave pública juntamente com as credenciais do Votante (conhecidas pelo Promotor), devolva um token assinado pelo Promotor (utilizando o seu certificado de assinatura) a autorizar a realização do voto.



Voto Electrónico 1 – período de votação

Passo 1:

- O Votante acede ao interface web ou app de votação, cujo endereço lhe foi comunicado pelo Promotor;
- O Votante introduz as credenciais pedidas (umas conhecidas pelo votante e pelo SMV e outras conhecidas pelo votante e pelo Promotor);
- É gerado um par de chaves (*prv_v*, *pub_v*) e um número aleatório *na* para este Votante.



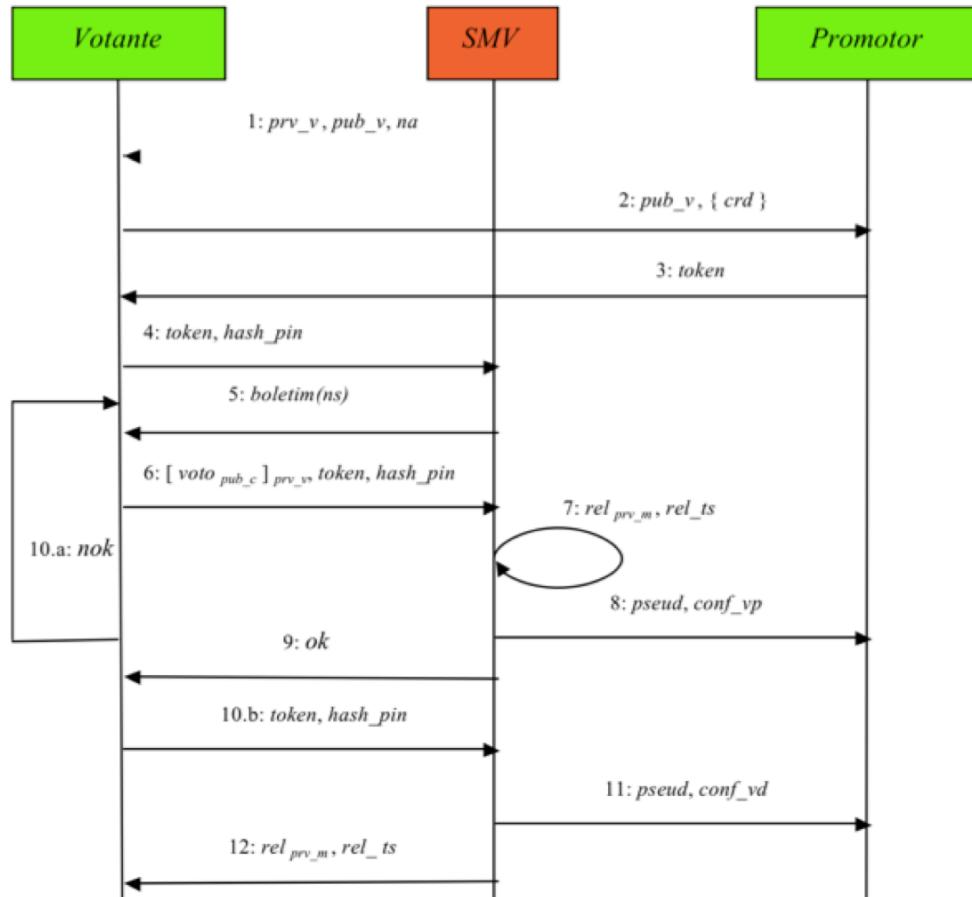
Passo 2:

- O interface web ou app de votação efectua (sem intervenção do Votante) um POST HTTPS para o serviço Web disponibilizada pelo Promotor (validando o serviço pelo certificado de servidor Web), enviando a chave pública gerada (*pub_v*) e as credenciais (conhecidas pelo Votante e pelo Promotor) introduzidas no passo anterior.

Voto Electrónico 1 – período de votação

Passo 3:

- a. Caso as credenciais recebidas estejam correctas, o Promotor devolve um token assinado por si (com a chave privada de assinatura) que autoriza o Votante a participar na eleição. Deste token é possível extrair o pseudónimo do Votante, o grupo a que pertence o Votante e quais os boletins de voto a que deverá ter acesso, e o número de votos que a sua votação representará, assim como contém a chave pública do votante (*pub_v*).



Passo 4:

- a. O interface web ou app de votação envia (sem intervenção do Votante) o token de votação (recebido do Promotor) para o SMV, juntamente com a hash das credenciais conhecidas entre Votante e SMV (*hash_pin*).

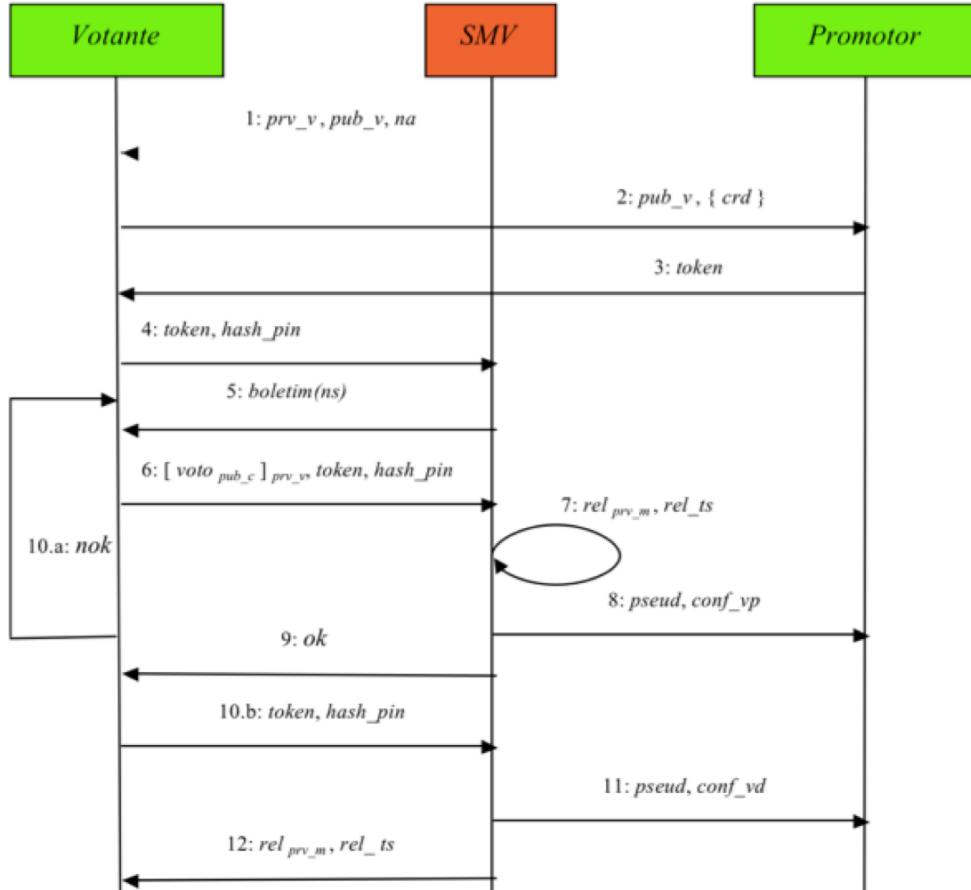
Voto Electrónico 1 – período de votação

Passo 5:

- O SMV indica à interface web/app de votação quais os boletins que o Votante está autorizado a visualizar e preencher;
- O Votante efectua a votação de acordo com a política de cada boletim de Voto.

Passo 6:

- A interface web/app de votação cria o voto que, para além das opções seleccionadas, inclui também o número aleatório *na* gerado anteriormente, cifrando-o com a chave pública do Promotor (*pub_c*, obtida do certificado de votação do Promotor) e assinando-o com a chave privada *prv_v* do Votante, gerada no primeiro passo.
- A assinatura gerada é enviada ao SMV, juntamente com o token e com a *hash_pin*.



Voto Electrónico 1 – período de votação

Passo 7:

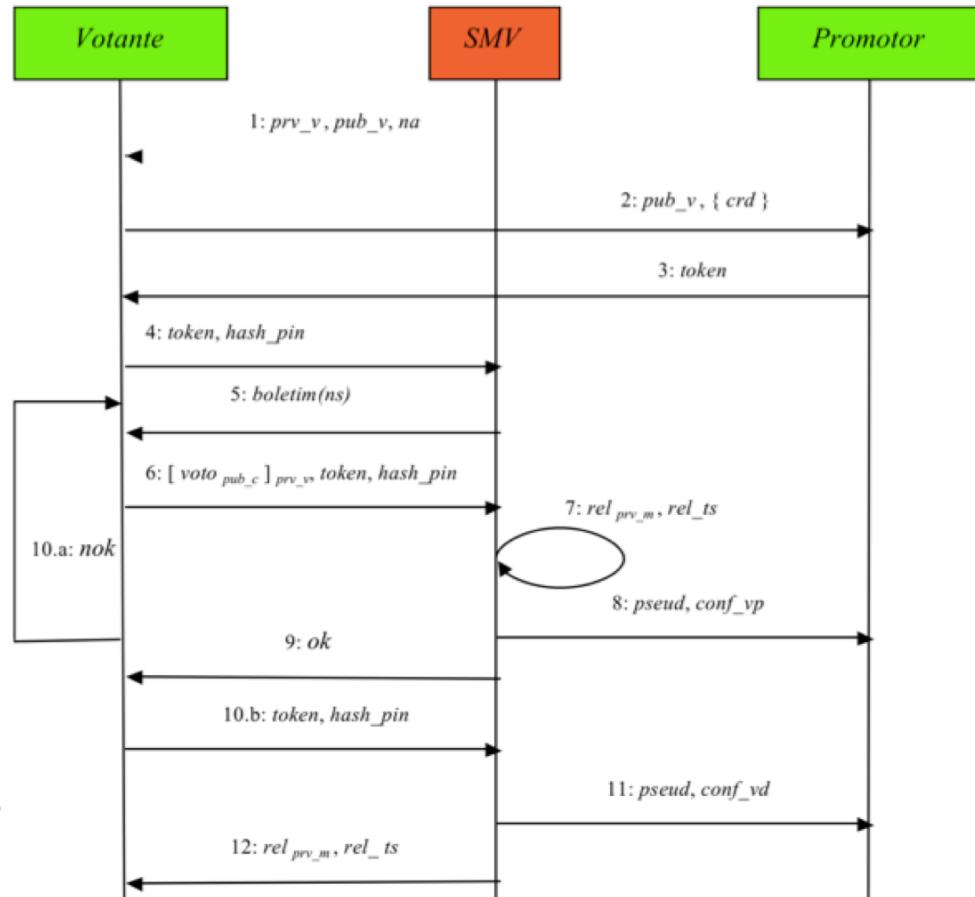
a. O SMV verifica se:

- reconhece o *hash_pin* (*hash_pin* gerados são únicos)
- o token é válido e,
- o voto foi assinado com a chave privada (*do votante*) que corresponde à chave pública contida no token.

b. O SMV gera um relatório de recepção onde inclui o pseudónimo do Votante *pseud*, assinando-o com a chave privada *prv_m* do SMV, dando origem a *rel_{prv_m}*.

c. O SMV obtém um timestamp sobre o relatório assinado, por forma a ter a prova da data/hora legal a que terminou de processar o recebimento do voto, ficando com *rel_ts*.

d. Se a data/hora que consta de *rel_ts* se encontrar fora do período da votação, o voto é descartado e é devolvida uma mensagem de erro no passo 9. Caso contrário, o SMV considera o voto válido e armazena-o na sua base de dados, juntamente com o relatório de recepção e respectivo timestamp.



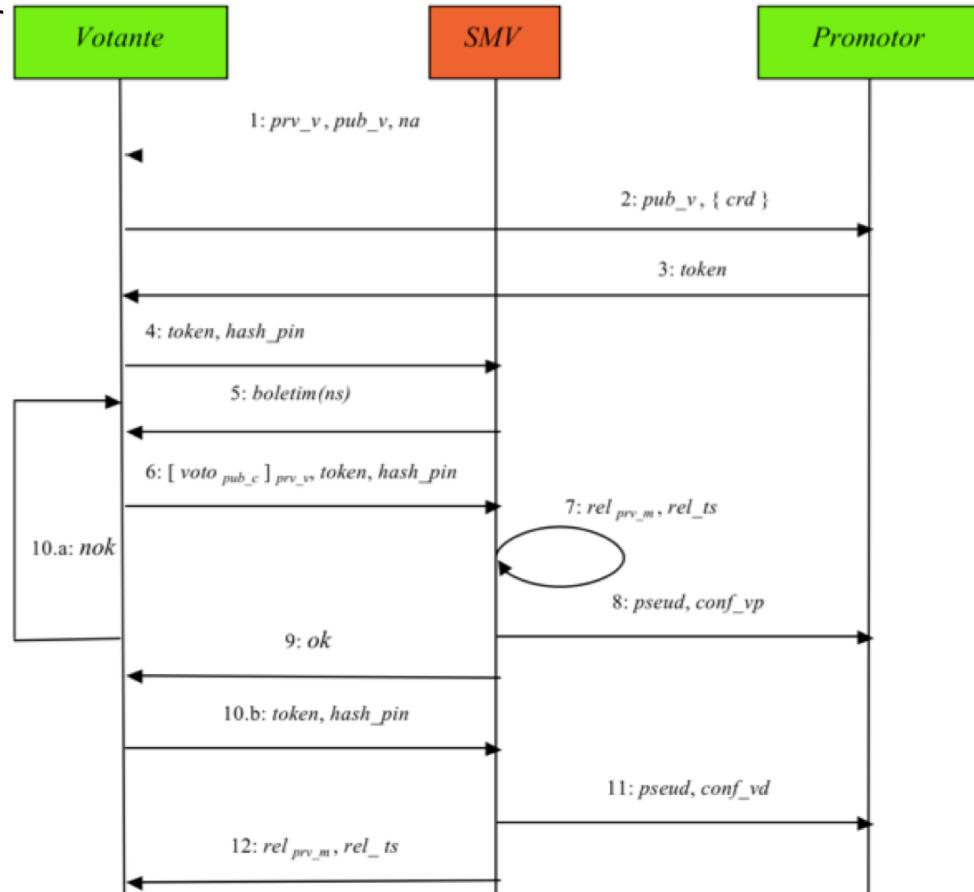
Voto Electrónico 1 – período de votação

Passo 8:

- a. Se as verificações efectuadas no passo anterior forem bem sucedidas é enviado, por POST HTTPS para um serviço Web do Promotor, uma confirmação provisória *conf_vp* de que o Votante com o pseudónimo *pseud* acabou de exercer o seu direito de voto.

Passo 9:

- a. Se as verificações efectuadas no passo 7 não tiverem sido bem sucedidas, é apresentada uma mensagem de erro explicativa ao Votante. Caso contrário, ser-lhe-ão mostradas as escolhas que fez em cada um dos boletins de voto, perguntando-lhe simultaneamente se as deseja confirmar.



Voto Electrónico 1 – período de votação

Passo 10a:

- Caso o Votante pretenda alterar o seu sentido de voto, o processo volta para o passo 5.

Passo 10b:

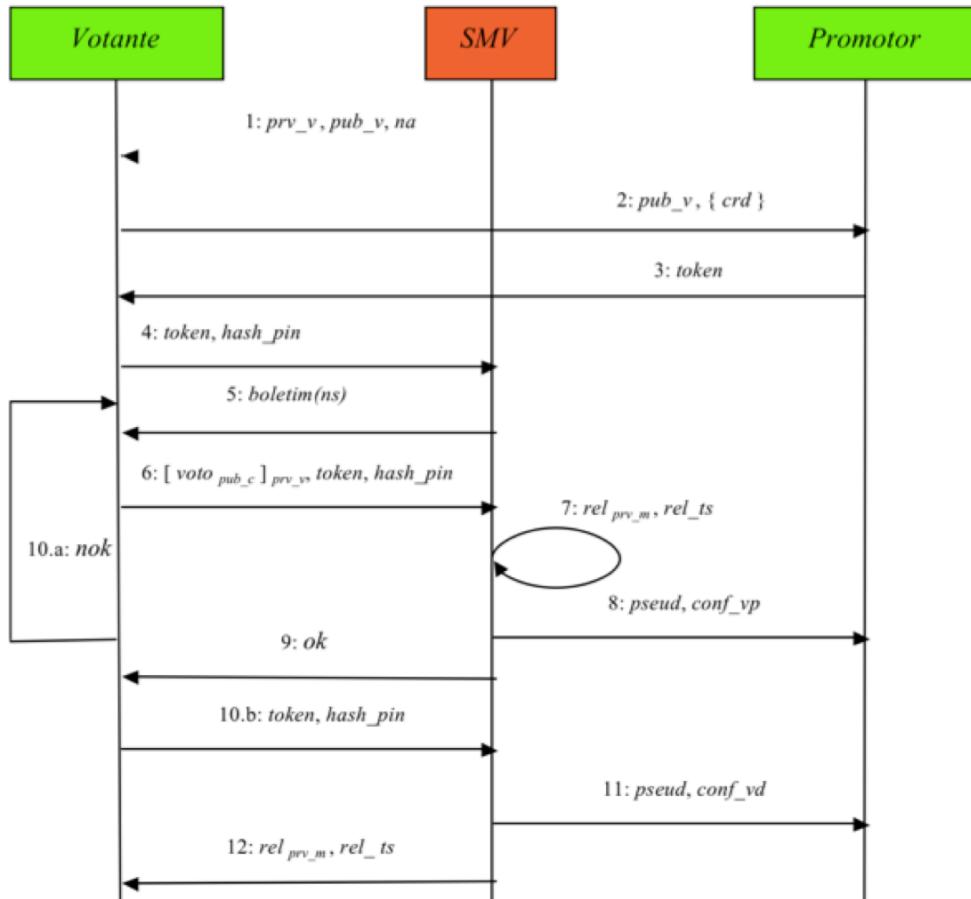
- Caso o Votante pretenda confirmar o seu sentido de voto, o token e a *hash_pin* do Votante são enviados novamente ao SMV, dando o voto como definitivo.

Passo 11:

- O SMV repete o passo 8 só que desta vez é enviada, ao Promotor, uma confirmação definitiva (*conf_vd*) do voto efectuado pelo Votante *pseud*.

Passo 12:

- O relatório de recepção assinado (rel_{prv_m}) e o respectivo time-stamp (rel_ts) gerados no passo 7 são devolvidos ao Votante para que este os possa armazenar como prova (com validade legal) de que exerceu o seu direito de voto dentro do período definido, sem que seja quebrada a confidencialidade do seu voto.



Voto Electrónico – Exemplo 1

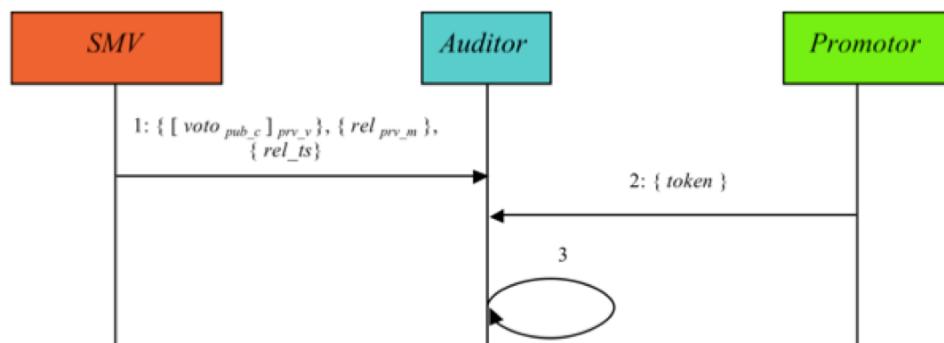
- Voto electrónico (fase pós-voto):
 - Findo o período de votação, o SMV envia ao Promotor:
 - conjunto de votos (cifrados mas sem a assinatura dos Votantes) assinados pelo SMV – garantindo que o Promotor não sabe quem votou o quê -;
 - relatórios de recepção de votos e os respectivos time-stamps obtidos.
 - O Promotor, com a sua chave privada de votação decifra os votos e efectua a contagem automática dos resultados da votação.
- Com os dados fornecidos pelo SMV, o Promotor pode:
 - comprovar se o número de votos que o SMV entregou está correcto (comparando com o número de autorizações de voto que emitiu),
 - saber o resultado da votação (decifrando os votos com a sua chave privada), e
 - ter a garantia que os votos recebidos deram entrada durante o período da votação.

Voto Electrónico – Exemplo 1

- Auditabilidade
 - Para preservar a confidencialidade do processo de votação, o SMV não pode entregar os votos ao Promotor na sua forma assinada pelo Votante (já que o Promotor passaria a saber quem tinha votado o quê, comprometendo-se o segredo do voto),
 - O Promotor pode levantar suspeitas sobre a veracidade e originalidade dos votos entregues pelo SMV;
 - Necessária a existência de um Auditor que, em caso de dúvida, pode comprovar a idoneidade do SMV utilizando um processo que também não lhe permite saber quem votou o quê.

Voto Electrónico – Exemplo 1

- Auditabilidade – esquema de funcionamento



Passo 1:

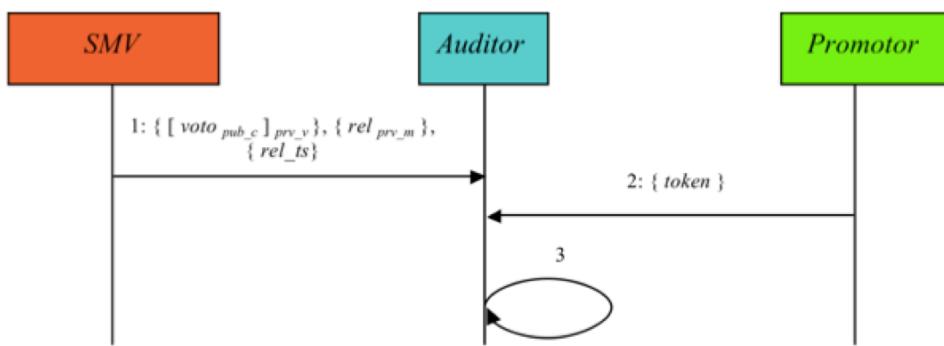
O SMV entrega ao Auditor os votos assinados que recebeu dos Votantes, os relatórios de recepção de votos assinados pelo SMV e os respectivos time-stamps.

Passo 2:

O Promotor entrega ao Auditor a lista dos tokens que emitiu durante o processo de votação.

Voto Electrónico – Exemplo 1

- Auditabilidade – esquema de funcionamento



Notas:

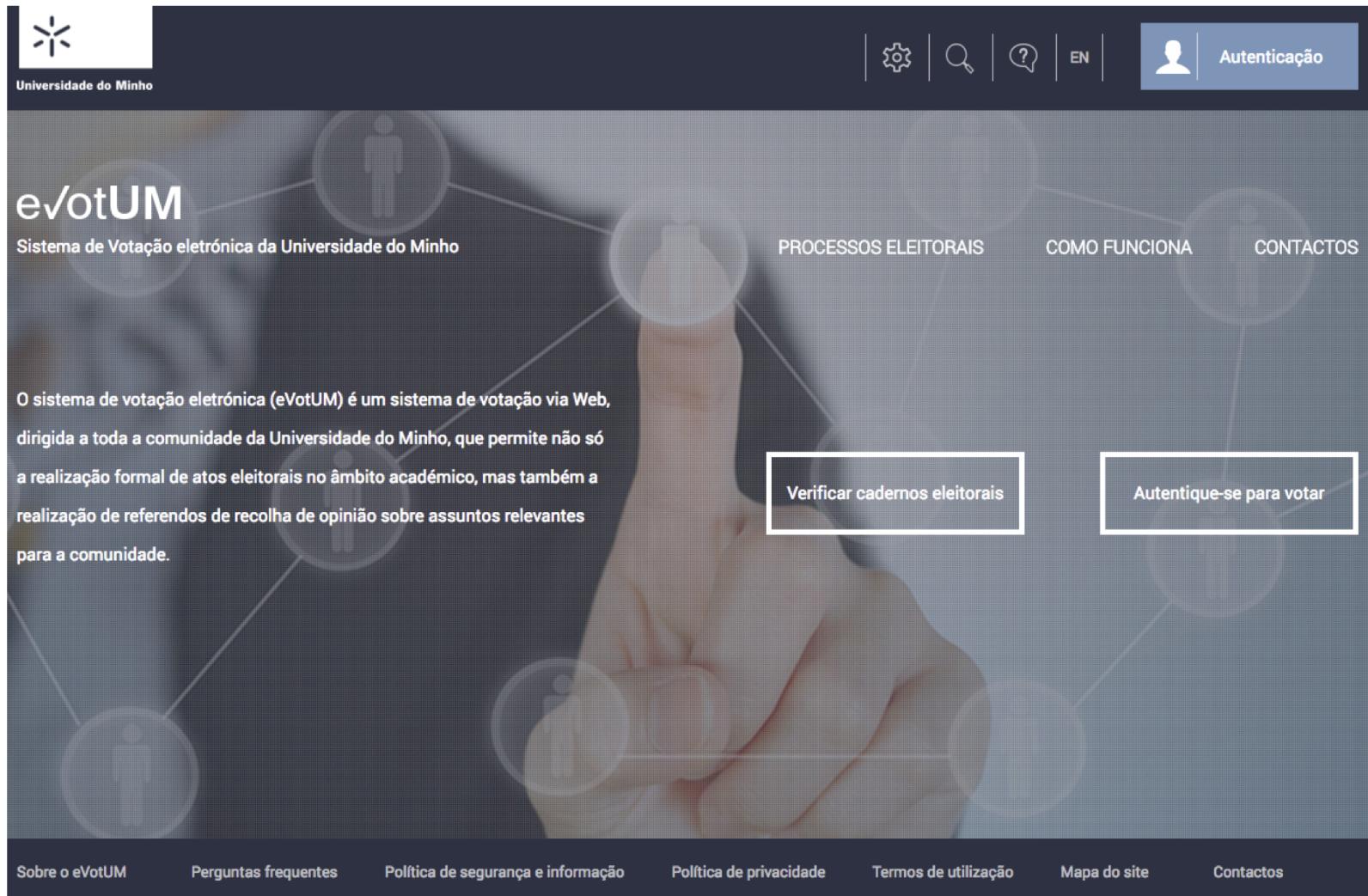
- Os formatos das chaves públicas, certificados, assinaturas digitais, mensagens cifradas e time-stamps utilizados pelo sistema são standard, pelo que o desenvolvimento d(a) aplicação(ões) a utilizar para proceder à auditoria dos dados fornecidos pode ser da responsabilidade do Auditor, garantindo-se deste modo a total independência da auditoria e a transparência de todo o processo de auditoria
- Durante este processo, o Auditor não conseguiu saber mais do que devia, já que o conteúdo dos votos assinados entregues pelo SMV se encontram cifrado para o Promotor, não possuindo o Auditor a chave privada necessária para os abrir.

Passo 3:

- O Auditor verifica se cada voto assinado entregue pelo SMV se encontra assinado com uma chave privada correspondente a uma chave pública contida num dos tokens entregues pelo Promotor.
- Adicionalmente, o Auditor pode verificar se a todos os votos entregues pelo SMV corresponde um relatório de recepção e respectivo time-stamp dentro do período definido para a votação.

Se tal acontecer, estará provado que o SMV não adulterou nenhum dos votos, não tendo por isso o Promotor razões para duvidar do resultado da votação.

Voto Electrónico – Exemplo 2



The screenshot shows the homepage of the eVotUM website. At the top left is the University of Minho logo. The top right features icons for settings, search, help, English version (EN), and authentication. The main header "eVotUM" is on the left, followed by the subtitle "Sistema de Votação eletrónica da Universidade do Minho". To the right are links for "PROCESSOS ELEITORAIS", "COMO FUNCIONA", and "CONTACTOS". A central image shows a hand interacting with a digital interface. Below the main content area are two prominent buttons: "Verificar cadernos eleitorais" and "Autentique-se para votar". At the bottom, there are links for "Sobre o eVotUM", "Perguntas frequentes", "Política de segurança e informação", "Política de privacidade", "Termos de utilização", "Mapa do site", and "Contactos".

Voto Electrónico – Exemplo 2

Características



AUTENTICIDADE

Apenas pessoas com direito a voto podem votar.



UNICIDADE

Cada eleitor vota apenas uma vez.



ANONIMATO

Não é possível associar um voto a um eleitor, nem vice-versa.



INTEGRIDADE

Os votos não podem ser modificados ou destruídos.



IRREVELÁVEL

Nenhum eleitor pode provar qual o voto que efetuou.



VERIFICABILIDADE

É possível verificar, de forma independente, que todos os votos foram contados corretamente.

Voto Electrónico – Exemplo 2

Características



AUDITABILIDADE

O sistema de voto eletrónico eVotUM pode ser testado e auditado por entidades independentes.



MOBILIDADE

O sistema de voto eletrónico eVotUM não restringe o local onde se vota.



TRANSPARÊNCIA

O sistema de voto eletrónico eVotUM é claro, exato, preciso e seguro.



DISPONIBILIDADE

O sistema de voto eletrónico eVotUM está sempre disponível durante o período de votação.



DETEÇÃO E RECUPERAÇÃO

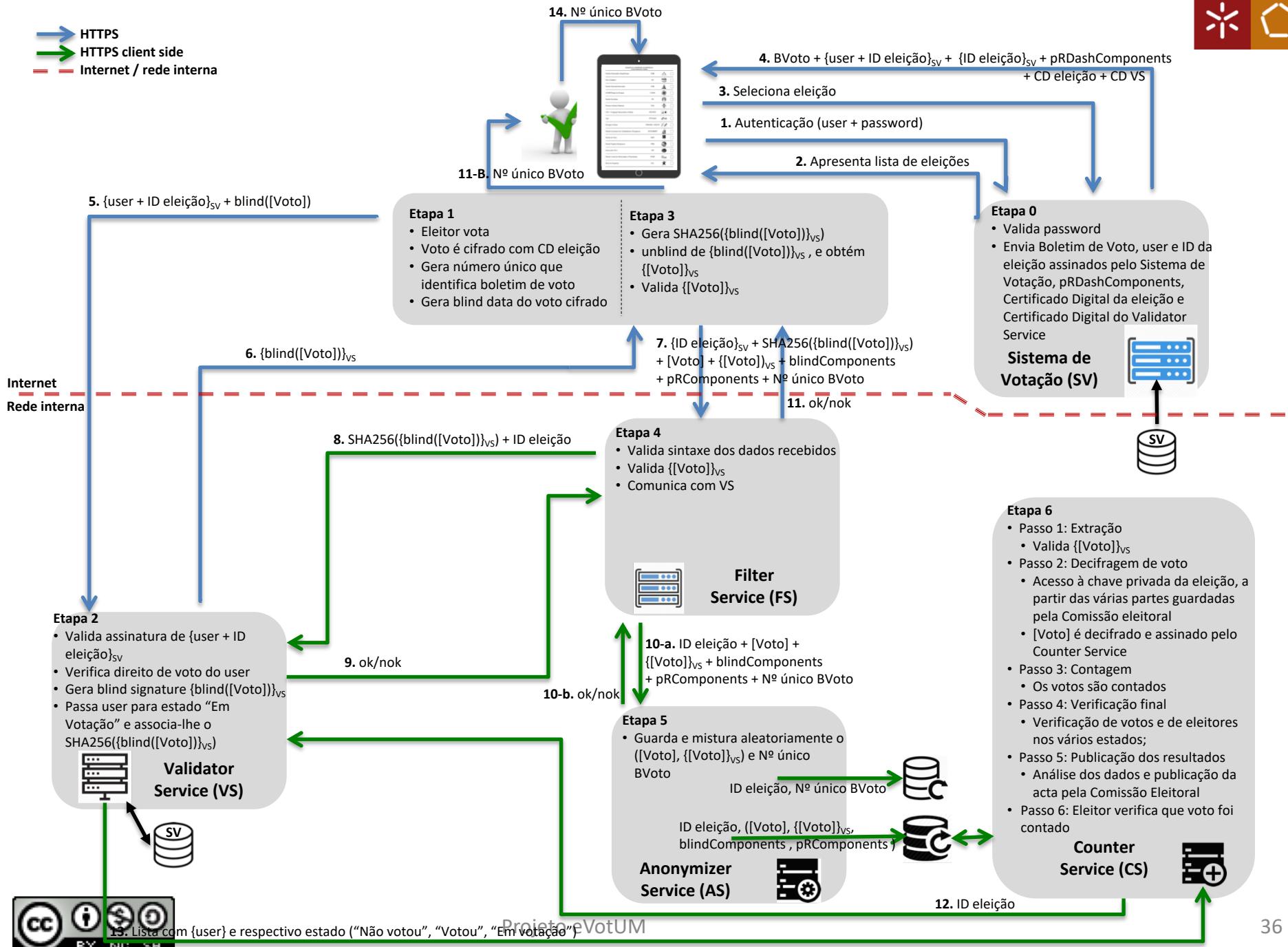
O sistema de voto eletrónico eVotUM deteta erros, falhas e ataques e, recupera a informação até ao ponto de falha.

Voto Electrónico – Exemplo 2

- Etapas e Fluxos de comunicação/mensagens



 HTTPS
 HTTPS client side
 Internet / rede interna



Eleição – Etapa 0

5. $\{\text{user} + \text{ID eleição}\}_{\text{SV}} + \text{blind}([\text{Voto}])$

11-B. N° único BVoto



14. N° único BVoto

4. BVoto + $\{\text{user} + \text{ID eleição}\}_{\text{SV}} + \{\text{ID eleição}\}_{\text{SV}} + \text{pRDashComponents}$
+ CD eleição + CD VS

3. Seleciona eleição

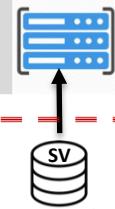
1. Autenticação (user + password)

2. Apresenta lista de eleições

Etapa 0

- Valida password
- Envia Boletim de Voto, user e ID da eleição assinados pelo Sistema de Votação, pRDashComponents, Certificado Digital da eleição e Certificado Digital do Validator Service

Sistema de Votação (SV)



- Eleitor acede a Sistema de Votação e autentica-se perante o mesmo;
- A autenticação é efectuada pelo envio de user + password do eleitor ao Sistema de Votação;
- O Sistema de Votação valida a password do eleitor (PBKDF2);
- O Sistema de Votação apresenta a lista de eleições nas quais o eleitor pode participar (eleições que estiverem no estado “aberta para votar”);
- O eleitor seleciona eleição;
- O Sistema de Votação envia ID da eleição e ($\text{user} + \text{ID da eleição}$) assinada com a chave privada do Sistema de Votação e, pRDashComponents, certificado digital da eleição e certificado digital do Validator Service.

Etapa 1

- Eleitor vota
- Voto é cifrado com CD eleição
- Gera número único que identifica batalha de voto

6. $\{\text{blind}([\text{Voto}])\}_{\text{VS}}$

Etapa 3

- Gera SHA256($\{\text{blind}([\text{Voto}])\}_{\text{VS}}$)
- unblind de $\{\text{blind}([\text{Voto}])\}_{\text{VS}}$, e obtém $[\text{Voto}]_{\text{VS}}$

7. $\{\text{ID eleição}\}_{\text{SV}} + \text{SHA256}(\{\text{blind}([\text{Voto}])\}_{\text{VS}} + [\text{Voto}] + [\text{Voto}]_{\text{VS}} + \text{blindComponents}$

11. ok/nok

• Valida sintaxe dos dados recebidos

8. $\text{SHA256}(\{\text{blind}([\text{Voto}])\}_{\text{VS}} + \text{ID eleição})$

Etapa 2

- Valida assinatura da eleição
- Verifica direito de voto do user
- Gera blind da eleição
- Passa user para estado “Em Votação” e gera SHA256($\{\text{blind}([\text{Voto}])\}_{\text{VS}}$)

9. ok/nok

Etapa 5

$([\text{Voto}], [\text{Voto}]_{\text{VS}})$ e N° único BVoto

ID eleição, $([\text{Voto}], [\text{Voto}]_{\text{VS}})$, blindComponents , pRDashComponents , Nº único BVoto

ID eleição, $([\text{Voto}], [\text{Voto}]_{\text{VS}})$, blindComponents , pRDashComponents , Nº único BVoto

Etapa 6

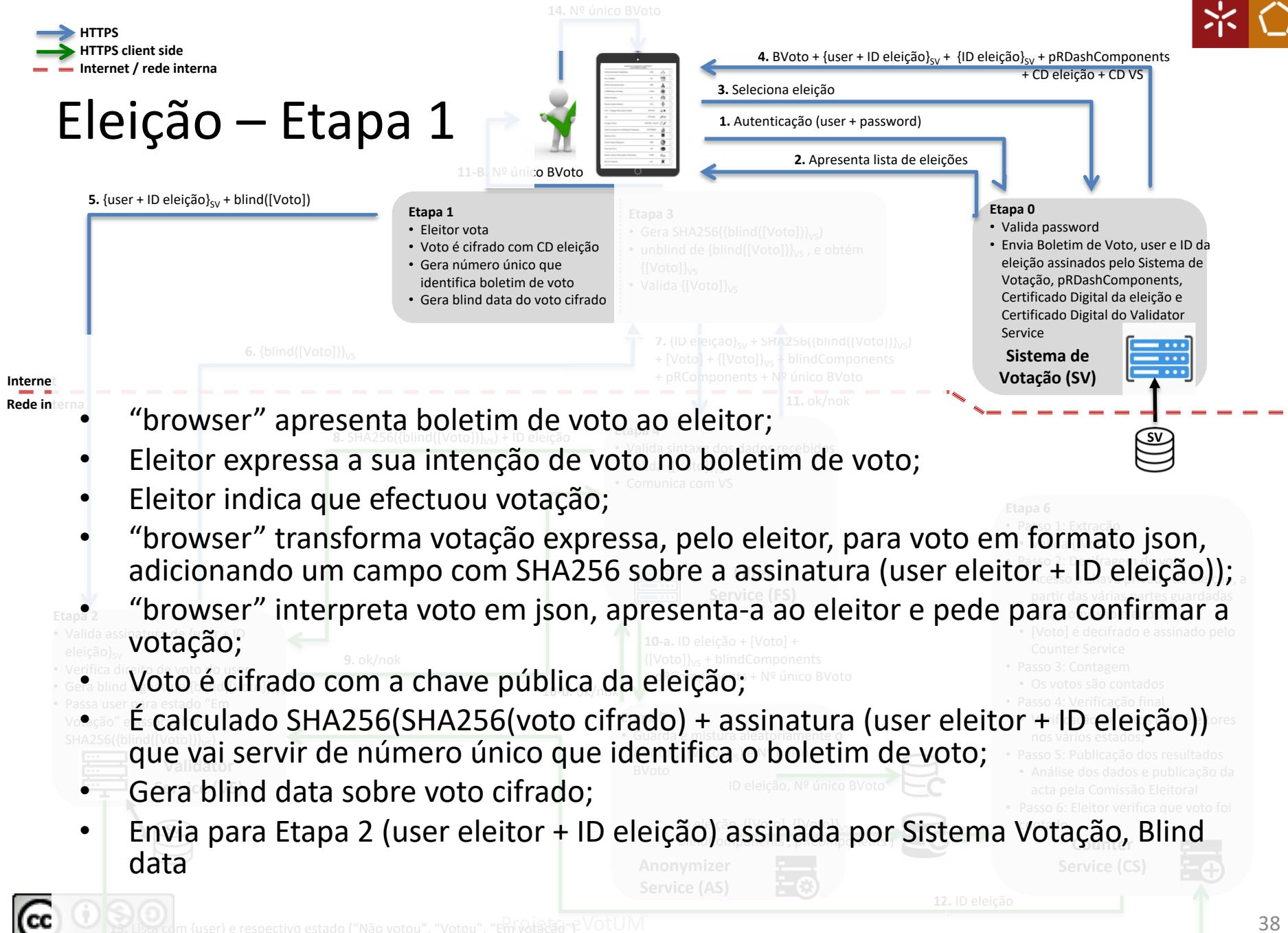
- Passo 1: Extração
 - Valida $[\text{Voto}]_{\text{VS}}$
- Passo 2: Decifragem de voto
 - Acesso à chave privada da eleição, a partir das várias partes guardadas pela Comissão eleitoral
 - $[\text{Voto}]$ é decifrado e assinado pelo Counter Service
- Passo 3: Contagem
 - Os votos são contados
- Passo 4: Verificação final
 - Verificação de votos e de eleitores nos vários estados;
- Passo 5: Publicação dos resultados
 - Análise dos dados e publicação da acta pela Comissão Eleitoral
- Passo 6: Eleitor verifica que voto foi contado

Counter Service (CS)

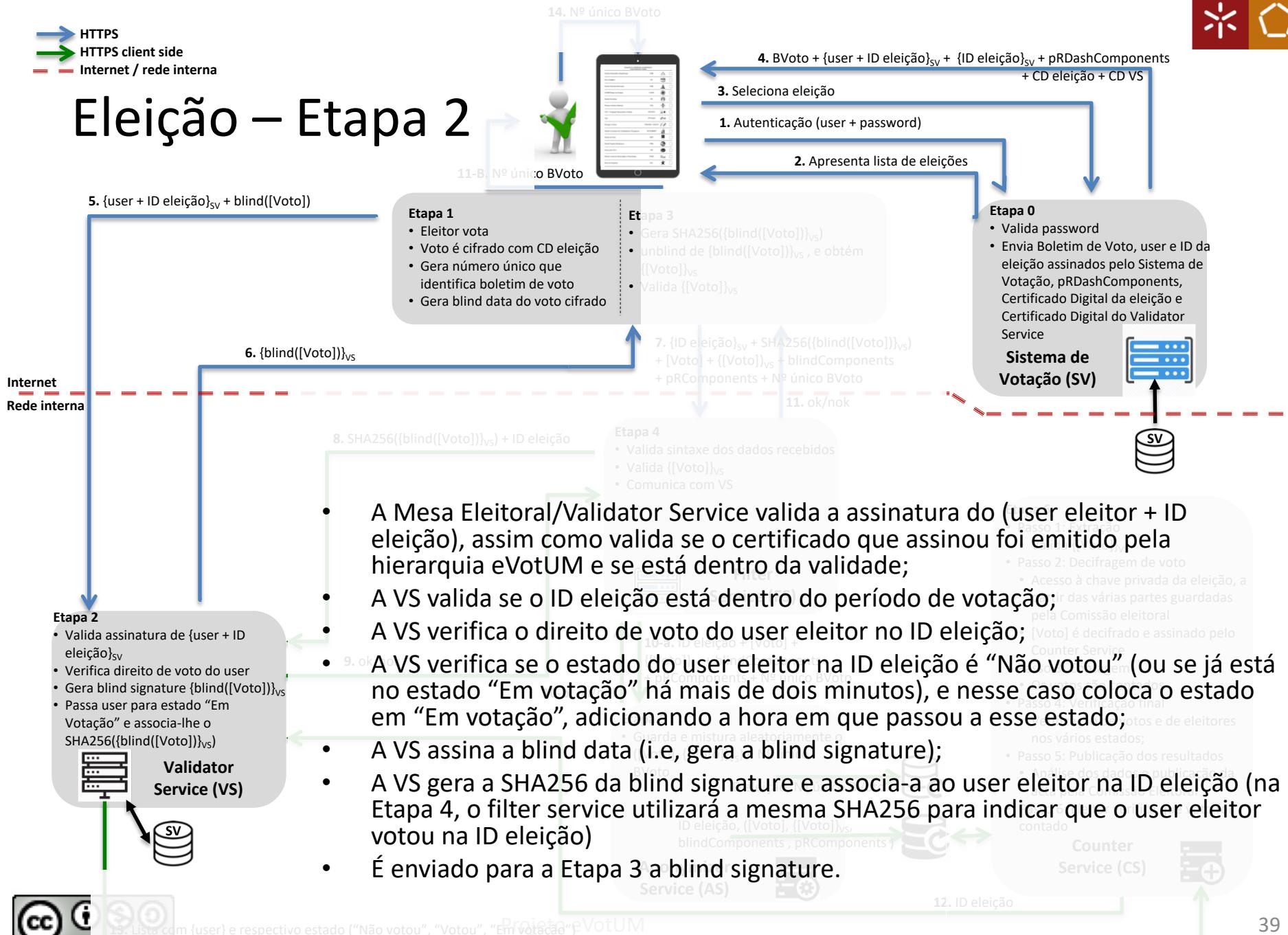


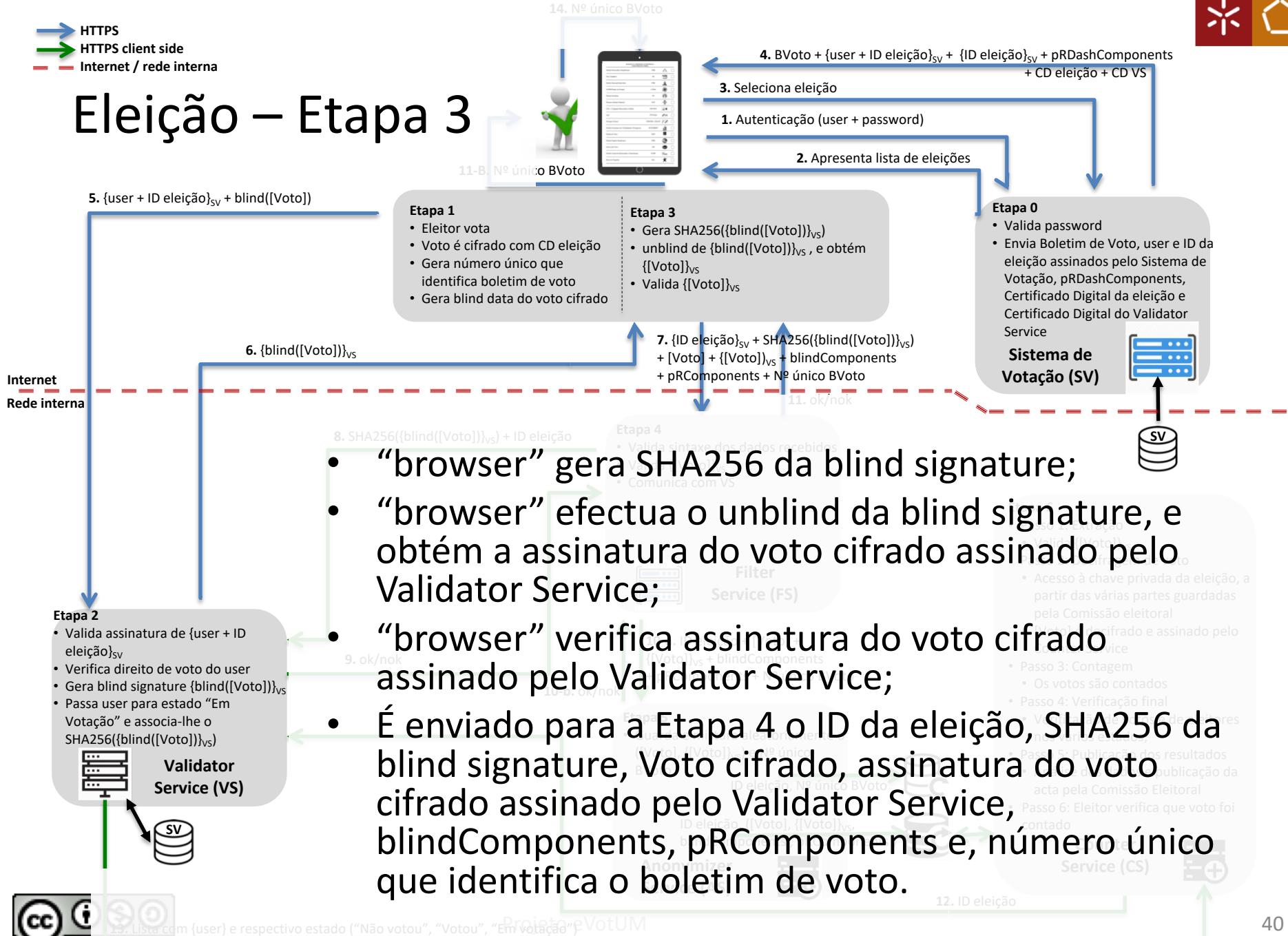
12. ID eleição

Eleição – Etapa 1

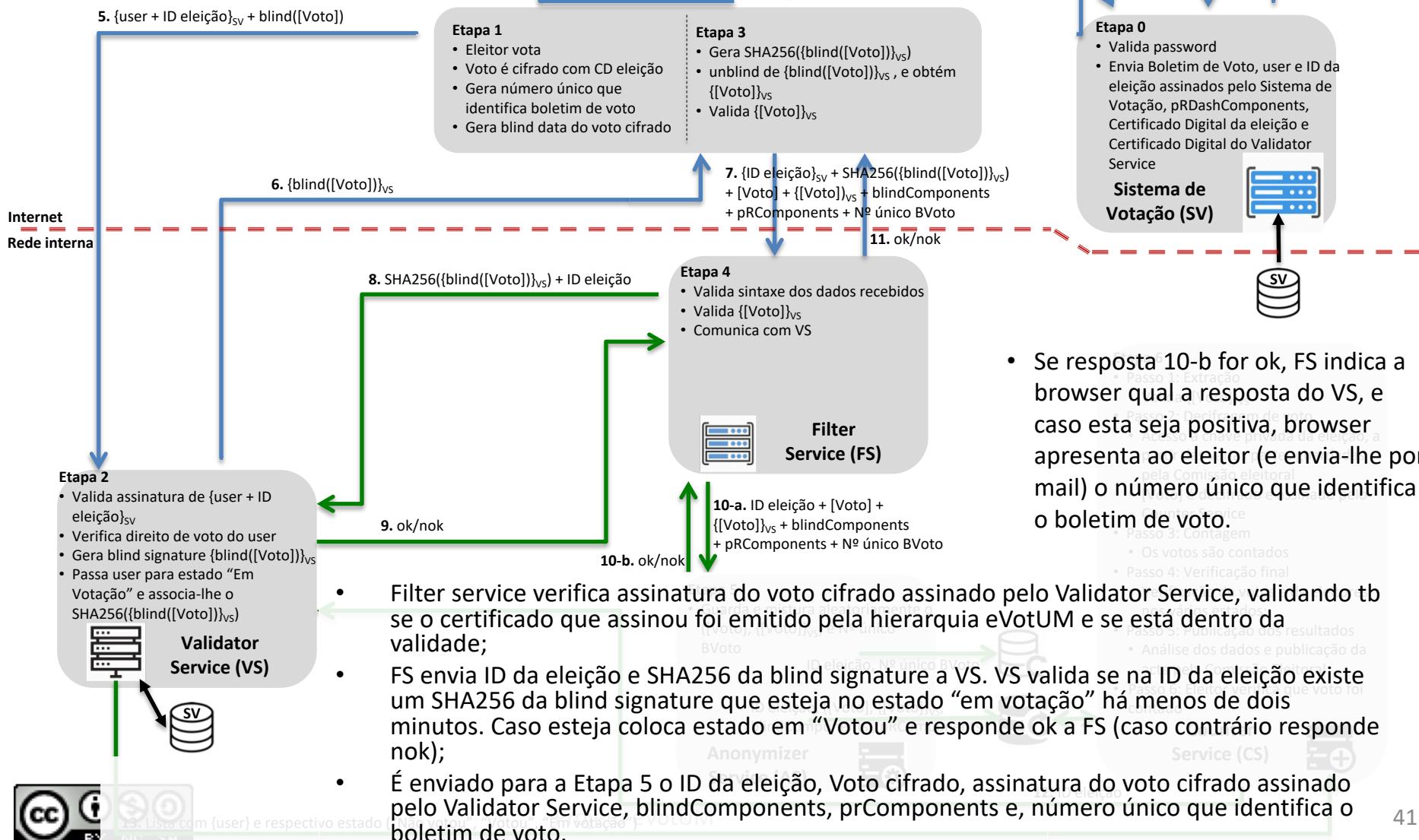


Eleição – Etapa 2





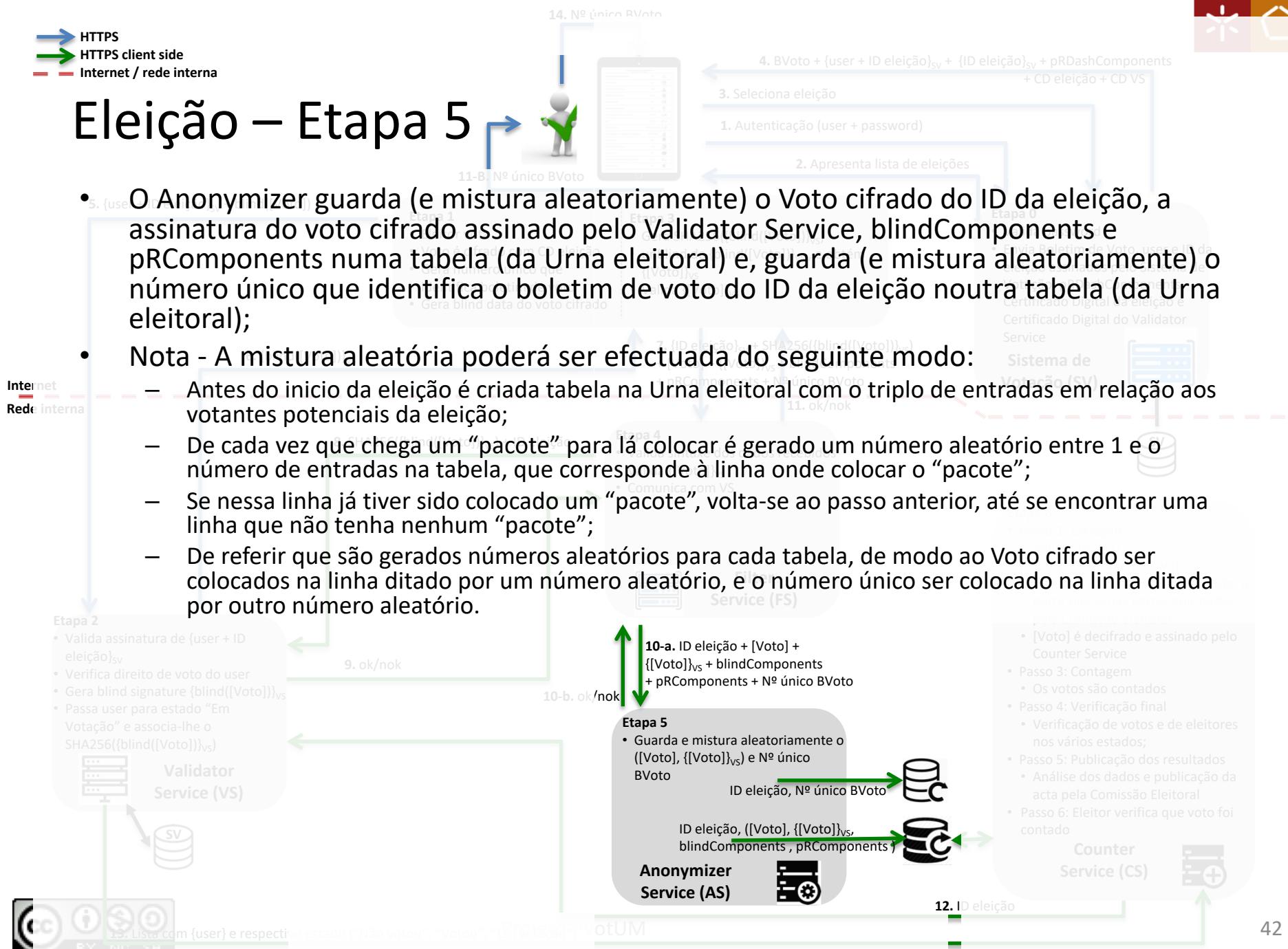
Eleição – Etapa 4





- HTTPS
- HTTPS client side
- Internet / rede interna

Eleição – Etapa 5



Eleição – Etapa 6

- Passo 5: Publicação dos resultados
- Passo 6: Eleitor pode verificar se o seu voto foi contado (através do número único que identifica o boletim de voto).

Passo 1: Extração

- Counter service verifica assinatura do voto cifrado assinado pelo Validator Service, validando tb se o certificado que assinou foi emitido pela hierarquia eVotUM e se está dentro da validade;

Passo 2: Decifragem de voto

- É gerada a password de acesso à chave privada da eleição, a partir das várias partes guardadas pela Comissão eleitoral;
- O voto é decifrado, validando-se a estrutura json do voto;
- O voto (com *timestamp*) é assinado pelo Counter Service, sendo guardado na BD;

Passo 3: Contagem

- Todos os votos são contados, a partir dos votos assinados;

Passo 4: Verificação final

- É pedido ao VS o número de eleitores nos vários estados (“Não votou”, “Votou”, “Em votação”) para a eleição;
- É indicado o número de votos recebidos, com assinatura do Validator Service correcta, decifrados correctamente, com a estrutura json do voto decifrado correcta

Etapa 0

- Valida password
- Gera SHA256([blind([Voto])]_{VS})
- Gera blind data do voto cifrado
- + [voto] + ([voto])_{VS} + blindComponents
- + pRComponents + Nº único BVoto
- 11. ok/nok

Votação (SV)



Etapa 6

- Passo 1: Extração
 - Valida {[Voto]}_{VS}
- Passo 2: Decifragem de voto
 - Acesso à chave privada da eleição, a partir das várias partes guardadas pela Comissão eleitoral
 - [Voto] é decifrado e assinado pelo Counter Service
- Passo 3: Contagem
 - Os votos são contados
- Passo 4: Verificação final
 - Verificação de votos e de eleitores nos vários estados;
- Passo 5: Publicação dos resultados
 - Análise dos dados e publicação da acta pela Comissão Eleitoral
- Passo 6: Eleitor verifica que voto foi contado

Counter Service (CS)

