Departamento de
Informática e Estatística
CTC • UFSC

Relatório Parcial

Compartilhamento Seguro de Segredos: Uma Implementação do Esquema de Shamir

INE5429 – Segurança em Computação

Alunas:
Jessica Regina dos Santos – 22100626
Myllena da Conceição Corrêa – 22104061

Florianópolis, SC

Sumário

1. Introdução

A segurança da informação tornou-se um dos pilares fundamentais da ciência da computação moderna, sobretudo em cenários que envolvem o armazenamento e a manipulação de dados sensíveis. Sistemas distribuídos, aplicações bancárias, carteiras digitais e ambientes corporativos críticos frequentemente demandam mecanismos robustos de proteção contra falhas, invasões e vazamentos. Dentro deste contexto, os esquemas de compartilhamento de segredos se destacam como uma solução matemática altamente eficaz.

Entre os diversos esquemas existentes, o Esquema de Compartilhamento de Segredos de Shamir, proposto por Adi Shamir em 1979, destaca-se por sua base teórica sólida sem depender de premissas criptográficas complexas. O funcionamento do esquema é baseado na álgebra sobre corpos finitos e no teorema da interpolação polinomial, possibilitando que um segredo seja dividido em n partes, das quais qualquer subconjunto com pelo menos t elementos possa reconstruí-lo — sendo que subconjuntos menores não revelam absolutamente nenhuma informação (Shamir, 1979; Boneh & Shoup, 2020).

A escolha deste algoritmo como objeto de estudo se deu justamente por sua relevância prática e por sua aplicabilidade em diversas áreas da segurança computacional. Em especial, sua simplicidade conceitual aliada à profundidade matemática oferece um excelente equilíbrio entre teoria e prática, permitindo a construção de soluções seguras (Stinson, 2005; Menezes et al., 1996).

Neste trabalho, exploraremos detalhadamente os fundamentos do esquema de Shamir, analisando seu funcionamento matemático, propriedades de segurança com um exemplo ilustrativo. A seguir, propomos um experimento de aplicação prática com uso simulado da técnica em um sistema de cadastro de usuário, visando demonstrar sua eficácia e adaptabilidade em ambientes computacionais reais.

2. Desenvolvimento

A segurança em computação é uma área essencial na ciência da computação. Dentro de cenários como sistemas distribuídos, armazenamento e aplicações financeiras, surgiu a necessidade de mecanismos que permitam a proteção de dados sensíveis, como chaves criptográficas ou informações confidenciais, diante da possibilidade de falhas, ataques e corrupção de dados. Uma abordagem robusta para esses problemas é o uso de esquemas de compartilhamento de segredos, que possibilitam a divisão de uma informação secreta entre diversos participantes, de forma que apenas um subconjunto mínimo deles seja capaz de reconstruí-la (Stallings, 2016).

Proposto por Adi Shamir em 1979, o Esquema de Shamir é um dos métodos mais reconhecidos na literatura utilizados para esse fim (Shamir, 1979). O esquema é do tipo (t, n) , onde um segredo é dividido em n partes, sendo necessárias ao menos t delas para recuperar a informação original. O diferencial desse método é sua base teórica sólida na matemática de corpos finitos e na interpolação polinomial, o que lhe confere propriedades valiosas como segurança perfeita e tolerância a falhas (Boneh & Shoup, 2020).

Ou seja, em vez de confiar em um único ponto de armazenamento ou autoridade para manter o segredo, o esquema de Shamir distribui os riscos entre múltiplos agentes. Esta característica é particularmente útil em ambientes nos quais a confiança precisa ser descentralizada, como cofres criptográficos e sistemas de custódia de ativos digitais (Menezes et al., 1996). Sua aplicação também se estende a protocolos modernos de segurança, como blockchain e carteiras multi-assinatura (Boneh & Shoup, 2020).

Nesta seção apresentaremos os fundamentos teóricos do esquema de Shamir, detalhando seu funcionamento matemático e discutindo suas propriedades de segurança, com base em obras clássicas e contemporâneas da área da criptografia. Também traremos um mini-exemplo prático para ilustrar sua aplicação.

2.1 Fundamentação Matemática para o Esquema de Shamir

O funcionamento do esquema de compartilhamento de segredos de Shamir está profundamente enraizado em dois conceitos fundamentais da matemática abstrata: álgebra dos corpos finitos e interpolação polinomial. Esses conceitos fornecem garantias formais necessárias para assegurar a segurança, a exatidão e a possibilidade de reconstrução do segredo, sem abrir mão da eficiência computacional.

2.1.1 Corpos Finitos

Um corpo finito é uma estrutura composta por um conjunto finito de elementos em que estão definidas duas operações: adição e multiplicação, obedecendo aos mesmos axiomas dos corpos numéricos tradicionais (como os reais ou os racionais): associatividade, comutatividade, distributividade, existência de identidade aditiva e multiplicativa, e existência de inversos (Stinson, 2005).

O corpo finito mais simples é o corpo primo F_p, definidos como o conjunto dos inteiros módulo p, onde p é um número primo. Ou seja:

$$F_p = \{0, 1, 2, \ldots, p - 1\}$$

Neste corpo, todas as operações são realizadas com aritmética modular, e cada elemento não nulo possui um inverso multiplicativo. Isso é essencial para a interpolação polinomial, que requer divisões (isto é, multiplicação por inversos). A escolha de *p* suficientemente grande é feita para acomodar o segredo e evitar colisões indesejadas durante os cálculos (Menezes et al., 1996).

Além disso, a existência de apenas um número finito de elementos garante que os polinômios definidos sobre F_p tenham propriedades específicas, como a unicidade da interpolação e a segurança contra reconstrução indevida com menos de *t* pontos.

2.1.2 Interpolação Polinomial

O segundo pilar matemático do esquema de Shamir é o teorema da interpolação polinomial, que afirma que, dados *t* pontos com abscissas distintas (x_1, y_1), (x_2, y_2), ..., (x_t, y_t), existe um único polinômio de grau no máximo $t - 1$ que passa por todos esses pontos. A forma mais comum de construir esse polinômio é por meio da interpolação de Lagrange.

A fórmula de Lagrange define o polinômio interpolador *f*(*x*), como:

$$f(x) = \sum_{j=1}^{t} y_j \cdot \ell_j(x)$$

Onde $\ell_j(x)$ é o polinômio base de La Grange, definido por:

$$\ell_j(x) = \prod_{1 \le m \le t,\, m \ne j} \frac{x - x_m}{x_j - x_m}$$

No contexto do esquema de Shamir, como se deseja recuperar apenas o segredo $S = f(0)$, a interpolação é feita especificamente em $x = 0$:

$$f(0) = \sum_{j=1}^{t} y_j \cdot \ell_j(0) = \sum_{j=1}^{t} y_j \cdot \prod_{1 \le m \le t,\, m \ne j} \frac{-x_m}{x_j - x_m}$$

Essas expressões são computadas em F_p, de modo que todas as divisões envolvem a multiplicação pelo inverso modular dos denominadores $(x_j - x_m) \, mod \, p$. Como F_p é um corpo, tais inversos sempre existem, garantindo a validade das operações (Stallings, 2016).

2.2 Funcionamento do Esquema de Shamir

O Esquema de Compartilhamento de Segredos de Shamir é um método criptográfico do tipo (*t*, *n*), ou seja, ele permite que um segredo seja dividido em *n* partes (ou *shares*), de modo que qualquer subconjunto com ao menos *t* dessas partes seja suficiente para reconstruir o segredo, enquanto subconjuntos com menos de *t* partes não fornecem qualquer informação sobre ele. O funcionamento matemático desse esquema está fundamentado em dois pilares: a aritmética de corpos finitos e a interpolação polinomial, particularmente pela fórmula de Lagrange, introduzidos na seção 2.1 (Shamir, 1979; Menezes et al., 1996).

2.2.1 Representação do segredo como polinômio

O segredo *S* é representado como o termo constante de um polinômio *f*(*x*) de grau $t - 1$, com coeficientes no corpo finito F_p, onde *p* é um número primo maior que *S* e os demais coeficientes. A escolha de F_p assegura que todas as operações realizadas, como adições, multiplicações e inversões, são fechadas e válidas no conjunto, além de garantir a existência de inversos multiplicativos necessários para a interpolação (Stallings, 2016).

O polinômio gerado aleatoriamente é da forma:

$$f(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_{t-1} x^{t-1}$$

Onde:

- $a_0 = S$ é o segredo que se deseja proteger;
- $a_1 + a_2 + \dots + a_{t-1} \in F_p$, são coeficientes aleatórios escolhidos uniformemente em F_p;
- $f(x) \in F_p[x]$ é um polinômio de grau $t - 1$.

Esse polinômio define uma função secreta cujos valores serão usados como os compartilhamentos individuais do nosso segredo.

2.2.2 Geração dos compartilhamentos

Para distribuir o segredo, o emissor escolhe n valores distintos $x1, x2, \dots, x_n \in F_p$ (geralmente os inteiros $1, 2, \dots, n$) e computa os pontos:

$$(y_j = f(x_j)) \text{ para } i = 1, 2, ..., n$$

Cada participante recebe o par (x_j, y_j), conhecido como seu compartilhamento (*share*). A segurança do esquema depende do fato de que, com menos de t desses pares, o polinômio $f(x)$ continua indeterminado, ou seja, existem infinitos polinômios compatíveis com os dados disponíveis, todos com valores distintos para $f(0) = S$.

Esse processo garante o que Shannon (1949) chamou de sigilo perfeito (*perfect secrecy*), pois a entropia do segredo permanece inalterada com base em $t - 1$ ou menos compartilhamentos (Boneh & Shoup, 2020).

2.2.3 Reconstrução do segredo via interpolação de Lagrange

E como reconstruiremos nosso segredo? Para recuperar o segredo, é necessário reunir pelo menos t compartilhamentos (x_j, y_j). Utiliza-se então a interpolação de Lagrange para reconstruir o polinômio $f(x)$, e em especial o valor $f(0) = a_0 = S$ (Menezes et al., 1996; Stinson, 2005).

A fórmula de Lagrange para reconstrução do valor $f(0)$ é dada por:

$$f(x) = \sum_{j=1}^{t} y_j \cdot \ell_j(x)$$

Onde $\ell_j(x)$ é o polinômio base de La Grange para o ponto x_j, definido por:

$$\ell_j(x) = \prod_{1 \leq m \leq t, m \neq j} \frac{x - x_m}{x_j - x_m}$$

E ao avaliar em $x = 0$, temos:

$$\ell_j(0) = \prod_{1 \leq m \leq t, m \neq j} \frac{-x_m}{x_j - x_m}$$

Todos os cálculos são realizados módulo p, o que exige o uso de inversos multiplicativos no corpo F_p. Como F_p é um corpo, esses inversos existem para qualquer número diferente de zero, garantindo que a interpolação seja sempre viável (Stallings, 2016).

A interpolação de Lagrange não apenas permite reconstruir o polinômio original como também garante sua unicidade, desde que o grau máximo do polinômio seja $t - 1$ e os x_j sejam distintos, como formalizado na próxima seção.

2.2.4. Justificativa da segurança e unicidade

A segurança do esquema repousa sobre duas garantias matemáticas:

1. Unicidade da interpolação polinomial: A interpolação polinomial em corpos finitos possui a propriedade de que, dado um conjunto de t pares (x_j, y_j) com x_j distintos, existe um único polinômio de grau menor que t que satisfaz $f(x_j) = y_j$ para todo j. Essa unicidade é garantida pelas propriedades dos espaços vetoriais sobre corpos, onde um sistema de t equações lineares em t incógnitas (os coeficientes do polinômio) tem solução única (Stinson, 2005; Menezes et al., 1996).

2. Segurança perfeita com menos de t pontos: Se menos de t compartilhamentos forem revelados, então o sistema de equações fica subdeterminado: há infinitos polinômios de grau $t - 1$ que interpolam os pontos conhecidos. Como cada um desses polinômios pode ter um valor diferente em $x = 0$, o segredo $f(0)$ permanece completamente indeterminado. Essa característica garante o que Shannon (1949) definiu como sigilo perfeito, pois a informação conhecida não reduz a incerteza sobre o segredo (Boneh & Shoup, 2020).

   Assim, a segurança do esquema não depende de suposições computacionais (como dificuldade de fatoração ou de logaritmos discretos), mas de propriedades matemáticas estritamente determinísticas da interpolação em corpos finitos.

2.3 Exemplo ilustrativo

Seja $p = 17$, $t = 3$, e o segredo $S = 5$. Escolhemos coeficientes aleatórios: $a_1 = 2$, $a_2 = 7$. O polinômio é:

$$f(x) = 5 + 2x + 7x^2 \, mod\, 17$$

Calculamos os compartilhamentos:

- $f(1) = 5 + 2(1) + 7(1)^2 = 14$

- $f(2) = 5 + 2(2) + 7(2)^2 = 37\, mod\, 17 = 3$

- $f(3) = 5 + 2(3) + 7(3)^2 = 74\, mod\, 17 = 6$

Com os pares $(1, 14),\ (2, 3)\ e\ (3, 6)$, pode-se usar a interpolação de Lagrange para reconstruir $f(0) = 5$, o segredo.

Agora, vamos fazer passo a passo a reconstrução do segredo $f(0)$ usando a interpolação de Lagrange com os pares $(1, 14),\ (2, 3)\ e\ (3, 6)$.

Sabemos que estamos em F_{17}, ou seja, todos os cálculos são feitos $mod\ 17$.

<u>Objetivo</u>

Queremos reconstruir o segredo:

$$f(0) = a_0 = \sum_{j=1}^{3} y_j \cdot \ell_j(0)$$

onde $\ell_j(0)$ são os polinômios base de Lagrange avaliados em 0, calculados conforme:

$$\ell_j(0) = \prod_{1 \le m \le t,\, m \ne j} \frac{-x_m}{x_j - x_m}\, mod\ 17$$

Como temos os pares:

- $(x_1, y_1) = (1,\ 14)$
- $(x_2, y_2) = (2, 3)$
- $(x_3, y_3) = (3,\ 6)$

Calculamos os coeficientes de Lagrange $\ell_1(0),\ \ell_2(0),\ \ell_3(0)$.

$$\ell_1(0) = \frac{-x_2}{x_1 - x_2} \cdot \frac{-x_3}{x_1 - x_3} = \frac{-2}{1-2} \cdot \frac{-3}{1-3} = \frac{-2}{-1} \cdot \frac{-3}{-2} = 2 \cdot \frac{3}{2}$$

Agora em F_{17}: $\frac{3}{2}\ mod\ 17 = 3 \cdot 2^{-1}\, mod\ 17$.

Precisamos calcular o inverso de $2\, mod\ 17$, ou seja, $2^{-1}\, mod\ 17$. Sabemos que $2 \cdot 9 = 18 \equiv 1\, mod\ 17$, então: $2^{-1} \equiv 9\, mod\ 17$.

Logo:

$$\ell_1(0) = 2 \cdot (3 \cdot 9) = 2 \cdot 27 = 54\, mod\, 17 = 3$$

Consequentemente:

$$\ell_2(0) = \frac{-x_1}{x_2 - x_1} \cdot \frac{-x_3}{x_2 - x_3} = \frac{-1}{2-1} \cdot \frac{-3}{2-3} = \frac{-1}{1} \cdot \frac{-3}{-1} = (-1) \cdot 3 = -3\, mod\, 17 = 14$$

$$\ell_3(0) = \frac{-x_1}{x_3 - x_1} \cdot \frac{-x_2}{x_3 - x_2} = \frac{-1}{3-1} \cdot \frac{-2}{3-2} = \frac{-1}{2} \cdot \frac{-2}{1} = \frac{(-1 \cdot -2)}{2} = \frac{2}{2} = 1$$

Calculando $f(0)$, onde:

$$f(0) = y_1 \cdot \ell_1(0) + y_2 \cdot \ell_2(0) + y_3 \cdot \ell_3(0) = 14 \cdot 3 + 3 \cdot 14 + 6 \cdot 1$$

$$f(0) = 42 + 42 + 6 = 90\, mod\, 17 = 90 - 5 \cdot 17 = 90 - 85 = 5.$$

Resultado final: $f(0) = 5\, mod\, 17$

Logo, o segredo reconstruído é $S = 5$.

3. Experimento

Para demonstrar a aplicação prática do Esquema de Compartilhamento de Segredos de Shamir, detalhado na seção de Desenvolvimento, propõe-se a concepção de um protótipo de sistema seguro para o gerenciamento de dados de usuários. O objetivo deste experimento é projetar uma aplicação que não apenas armazena informações sensíveis, como o Cadastro de Pessoas Físicas (CPF) e outros dados sigilosos, mas que também incorpora um mecanismo de contenção de desastres criptograficamente seguro.

O cenário proposto tem a ideia de simular um sistema que visa assegurar a integridade e a confidencialidade dos dados. Em caso de alguma violação de segurança de forma catastrófica, como uma invasão confirmada por um agente externo, é necessário um mecanismo de *fail-safe* (Wikipedia, 2024) para proteger os dados dos usuários, inutilizando-os para o invasor. Esta ação drástica, que pode ser a exclusão ou a cifragem irreversível do banco de dados, será controlada por uma única Chave Mestra de Contenção. Armazenar esta chave em um único local representaria um ponto único de falha crítica.

É neste ponto que o Esquema de Shamir se torna a solução central do experimento. A Chave Mestra de Contenção será o segredo a ser protegido. Em vez de ser armazenada diretamente, ela será dividida em n partes (*shares*) usando um esquema do tipo (t, n) (Shamir, 1979). Por exemplo, em um esquema $(2,\ 3)$, a chave seria dividida entre três Administradores de Segurança, sendo necessária a colaboração de, no mínimo, dois deles para reconstruir a chave e ativar o protocolo de contenção. Esta abordagem descentraliza a responsabilidade e previne que um único administrador comprometido possa acionar o mecanismo indevidamente.

Nesta seção, descreveremos a arquitetura e o fluxo de funcionamento desta aplicação. A descrição será feita por meio de Diagramas de Fluxo, que ilustram o processo de cadastro de usuários e a ativação do mecanismo de segurança.

3.1 Diagramas de Fluxo

3.1.1 Diagrama do Cadastro Usuário

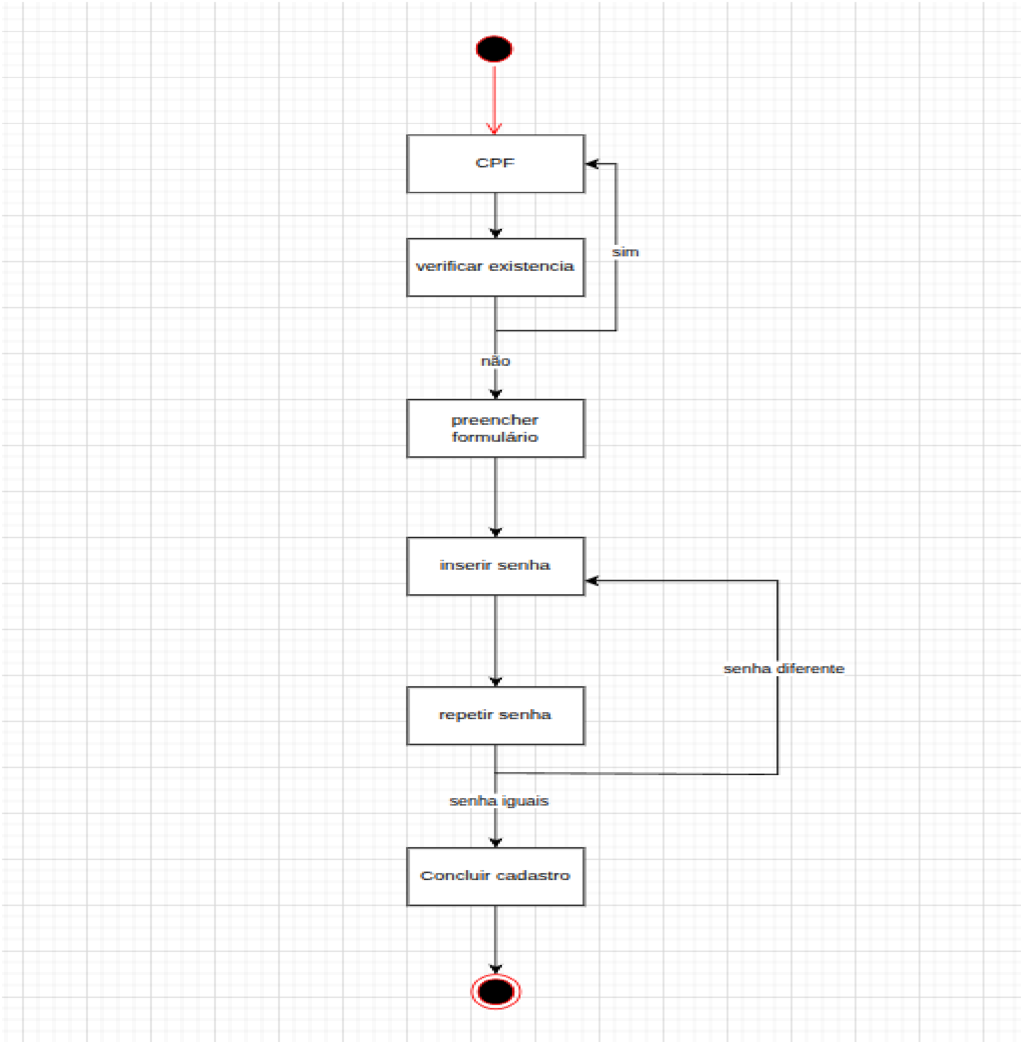


Imagem 1. Primeiro esquemático para o diagrama que descreve o processo básico de cadastro de usuário (rascunho).

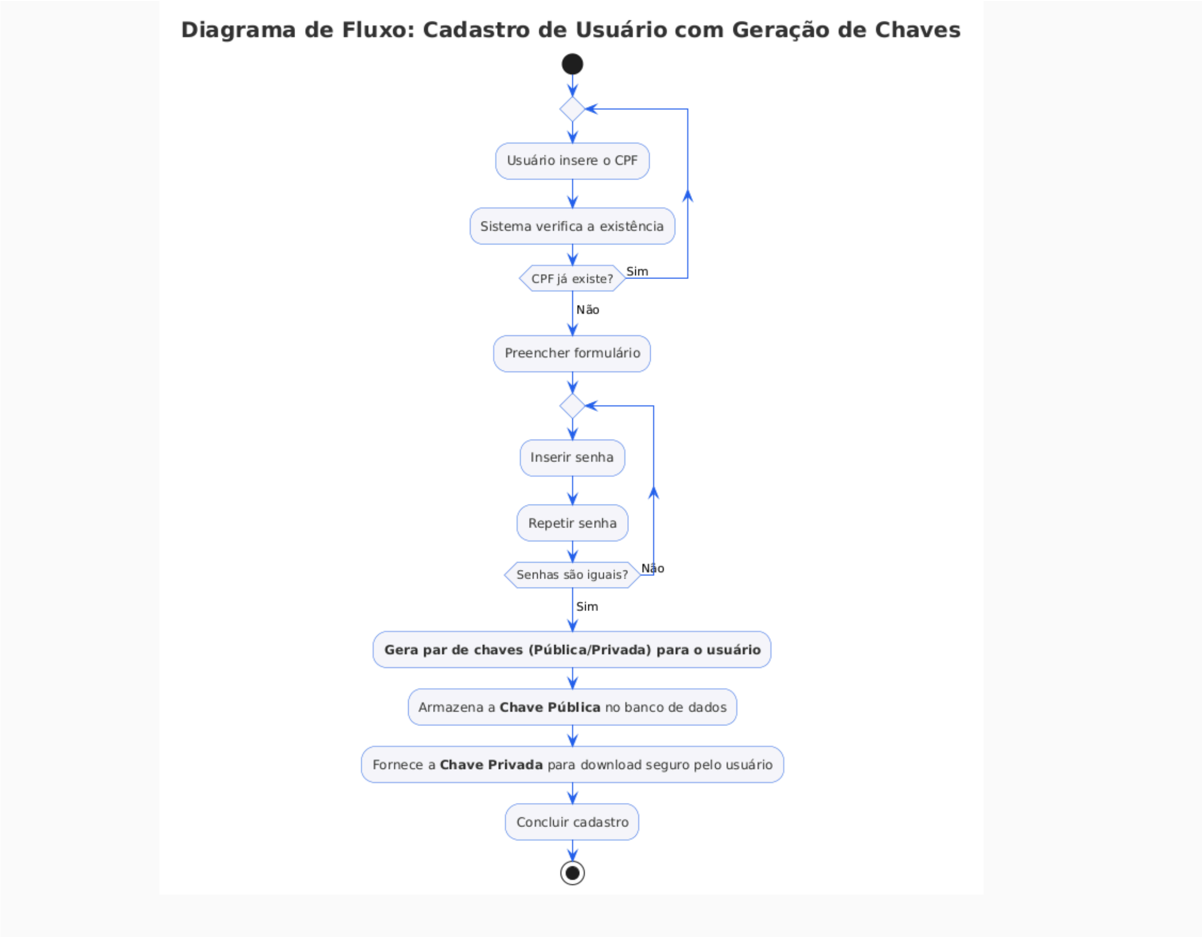


Imagem 2. Diagrama Final. Utilizamos a linguagem PlantUML para criação do diagrama.

- Início do Processo: O fluxo começa com a ação inicial de um usuário que deseja se cadastrar.
- Entrada do CPF: O primeiro dado solicitado ao usuário é o seu CPF, que servirá como identificador único no sistema.
- Verificação de Existência: Uma vez que o CPF é fornecido, o sistema realiza uma verificação para determinar se este identificador já existe em sua base de dados.
    - Se "sim" (o CPF já está cadastrado), o fluxo retorna ao passo anterior, solicitando novamente um CPF.
    - Se "não" (o CPF não está cadastrado), o processo continua para a próxima etapa.
- Preenchimento de Formulário: O usuário é então direcionado para preencher um formulário com informações adicionais de cadastro. Como nome, sobrenome, idade, etc.
- Criação de Senha: O sistema solicita que o usuário insira uma senha. Em seguida, é pedido que ele repita a senha para fins de confirmação.
- Validação da Senha: O sistema compara as duas senhas inseridas.
    - Se forem "diferentes", o fluxo retorna à etapa de "inserir senha", pedindo ao usuário que tente novamente.

    - Se forem "iguais", a validação é bem-sucedida e o processo avança.
- Geração de Chaves: O sistema gera um par de chaves criptográficas único para o usuário: uma pública e uma privada (Wikipedia, 2024).
- Armazenamento da Chave Pública: A chave pública, que serve para "trancar" mensagens para o usuário, é armazenada no banco de dados do sistema, associada ao seu perfil.
- Entrega da Chave Privada: A chave privada, que é secreta e serve para "abrir" as mensagens, é disponibilizada para download imediato pelo usuário e não é guardada pelo sistema. A responsabilidade de mantê-la segura é inteiramente do usuário (Wikipedia, 2024).
- Conclusão do Cadastro: Com todas as informações validadas, o sistema executa a rotina para "Concluir o cadastro", efetivamente criando a conta do usuário.
- Fim do Processo: O fluxo de cadastro é finalizado.

3.1.2 Diagrama de distribuição de chaves

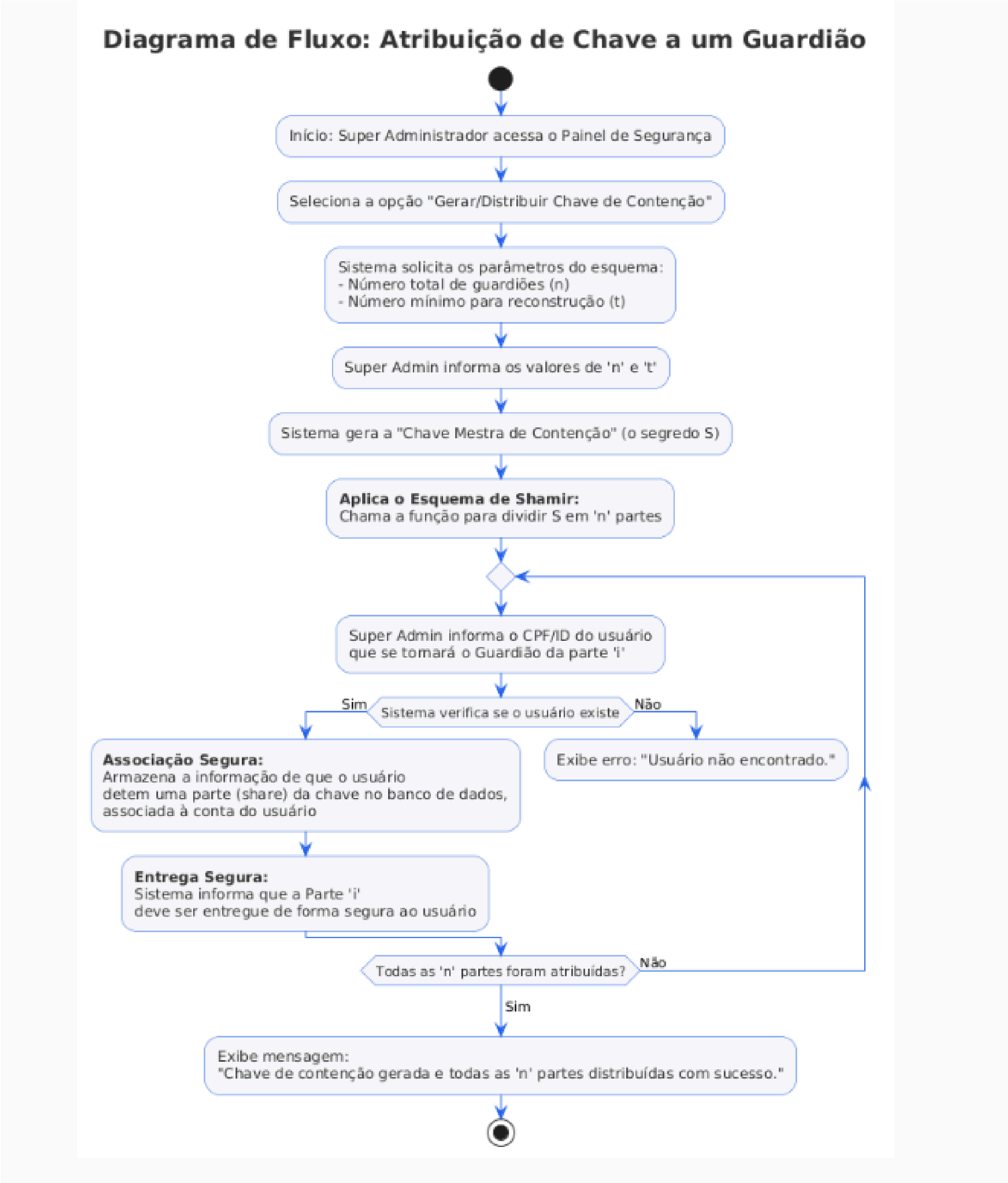


Imagem 3. Diagrama de distribuição de chaves criado através da linguagem PlantUML

Neste diagrama, o fluxo pode ser dividido nas seguintes fases:

- <u>Iniciação e Configuração</u>
  O processo começa quando o "Super Administrador" acessa o "Painel de Segurança". Após selecionar a opção para gerar a chave, o sistema solicita os

parâmetros fundamentais do esquema: n (o número total de “Guardiões”) e t (o número mínimo necessário para reconstrução). O “Administrador” insere esses valores, definindo as regras de segurança.

- Geração do Segredo e das Partes
  O sistema gera uma "Chave Mestra de Contenção" aleatória (S). Imediatamente, o algoritmo de Shamir é aplicado para dividir matematicamente o segredo S em n partes únicas (as *shares*). A chave mestra original pode ser descartada após este passo, pois o segredo agora existe de forma distribuída.

- Atribuição e Distribuição (Loop)
  O sistema entra em um loop que se repete n vezes. Em cada iteração, o “Super Administrador” informa o identificador (CPF/ID) do usuário que será o “Guardião” daquela parte. O sistema valida a existência do usuário e, em caso afirmativo, realiza a “Associação Segura”, armazenando a parte da chave no banco de dados de forma logicamente associada à conta daquele usuário para fins de auditoria.

- Entrega Segura e Conclusão
  A etapa final de cada loop é a “Entrega Segura”. Após todas as n partes terem sido atribuídas, o sistema exibe uma mensagem de sucesso e o processo é finalizado.

- Detalhamento da "Entrega Segura" com Criptografia Assimétrica
  A segurança de todo o protocolo depende da implementação da etapa de "Entrega Segura". A abordagem recomendada para garantir a confidencialidade durante a distribuição das partes da chave é o uso de Criptografia Assimétrica (ou de Chave Pública), utilizando algoritmos como RSA (Wikipedia, 2024). O processo ocorre da seguinte forma:

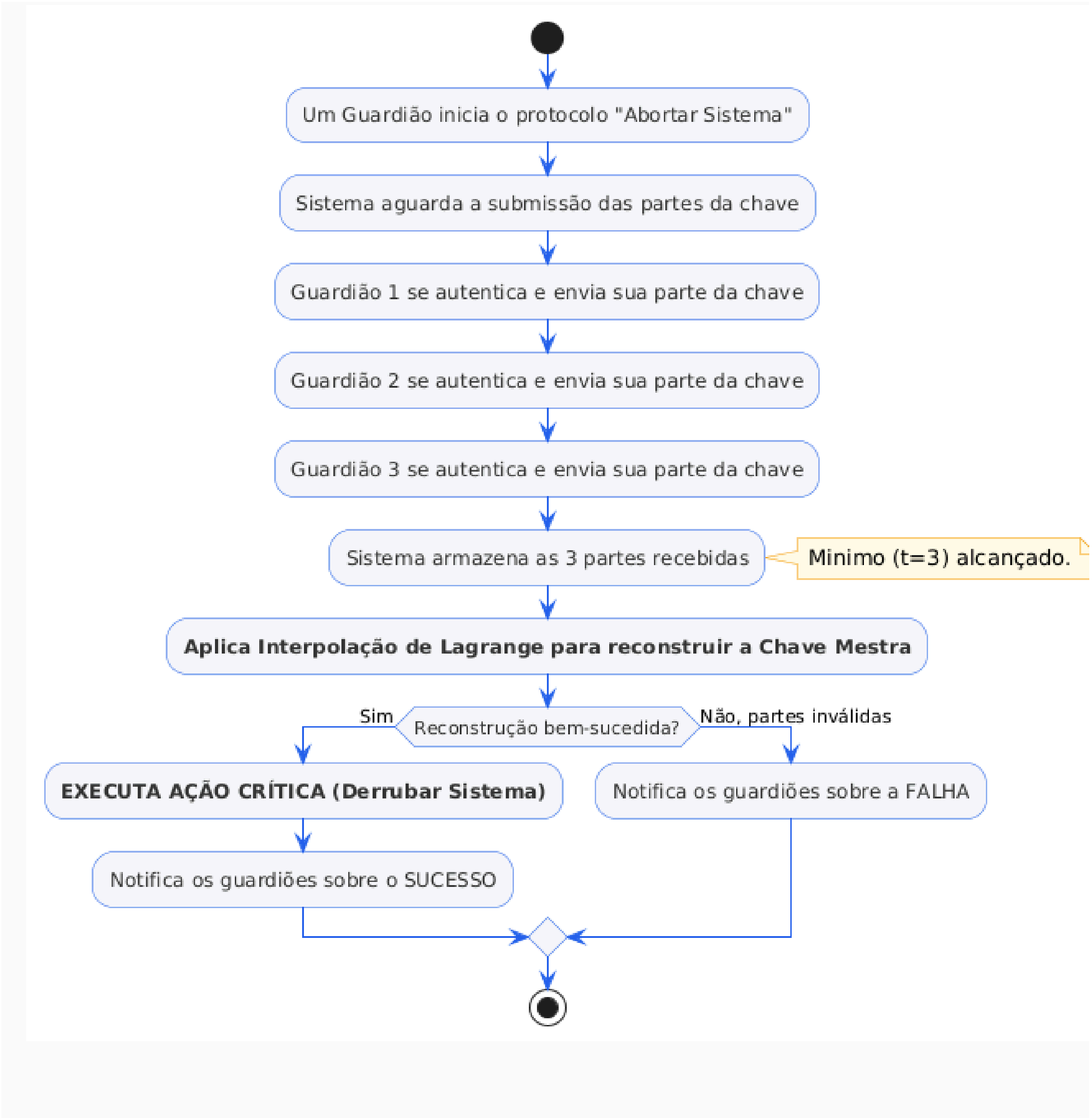
  - Pré-requisito: Cada usuário designado como “Guardião” deve possuir um par de chaves criptográficas: uma chave pública e uma chave privada. A chave pública de cada Guardião deve ser previamente registrada no sistema e associada à sua conta.

  - Cifragem: Quando o sistema precisa entregar a parte a um “Guardião”, ele primeiro recupera a chave pública daquele “Guardião” em seu banco de dados. Em seguida, o sistema utiliza essa chave pública para cifrar o conteúdo da parte.

  - Transmissão: A parte da chave, agora cifrada, pode ser transmitida por qualquer canal de comunicação. Mesmo que a mensagem seja interceptada, seu conteúdo permanecerá ilegível para qualquer pessoa que não possua a chave privada correspondente.

- Decifragem: Ao receber a sua parte cifrada, o "Guardião" utiliza a sua própria chave privada (que é secreta e conhecida apenas por ele) para decifrar a mensagem, revelando assim o conteúdo da sua parte da chave de forma segura. Este método garante que apenas o destinatário legítimo possa ter acesso à sua *share* (Wikipedia, 2024).

3.1.3 Diagrama para abortar o sistema



- Início
  O fluxo começa quando um dos “Guardiões" autorizados decide iniciar o protocolo "Abortar Sistema".

- Estado de Espera
  A ideia é o sistema entra em um modo de aguardo, esperando que os “Guardiões” submetam suas respectivas partes da chave (as *shares*) para autorizar a ação.

- Submissão das Partes
  Cada um dos três “Guardiões” designados deve se autenticar no sistema e enviar sua parte individual da chave. Essa submissão funciona como um voto de concordância para a execução do protocolo.

- Coleta e Verificação
  O sistema armazena as 3 partes recebidas. Na nota, o diagrama indica que neste momento o mínimo necessário para a decisão ($t = 3$) foi alcançado.

- Reconstrução da Chave Mestra
  O sistema aplica o algoritmo de Interpolação de Lagrange usando as 3 partes fornecidas para tentar recriar a "Chave Mestra" secreta original.

- Validação da Reconstrução
  O sistema avalia o resultado da etapa anterior.
    - Se "Sim" (Reconstrução bem-sucedida): significa que as partes eram válidas e a Chave Mestra foi recuperada com sucesso. O fluxo avança para a execução da ação final.
    - Se "Não" (Partes inválidas): significa que a reconstrução falhou, pois as partes fornecidas eram incorretas ou corrompidas. O fluxo desvia para o tratamento de falha.

- Execução da Ação Crítica
  Ocorre apenas no caminho "Sim". Com a Chave Mestra reconstruída e validada, o sistema executa a sua função mais drástica, a de "Derrubar o Sistema"
  .

- Notificação de Sucesso
  Após a execução da ação crítica, o sistema notifica os Guardiões de que o protocolo foi concluído com sucesso.

- Notificação de Falha
  Ocorre apenas no caminho "Não". O sistema informa aos Guardiões que a operação falhou e que a Chave Mestra não pôde ser reconstruída.

- Fim do Processo
  O fluxo de trabalho é encerrado, seja após a execução bem-sucedida ou após a notificação de falha.

3.2 Considerações finais

É importante ressaltar que os diagramas e descrições apresentados nesta seção foram criados visando fluxos de trabalho mais complexos e na lógica criptográfica.

Para manter a objetividade, funcionalidades mais básicas do funcionamento do sistema foram omitidas. O relatório descrito representa o projeto até esse momento, sendo possível que diagramas sejam criados ou ajustados durante o desenvolvimento.

Referências

Boneh, D., & Shoup, V. (2020). *A Graduate Course in Applied Cryptography*. Disponível em: https://toc.cryptobook.us

FAIL-safe. Wikipedia: The Free Encyclopedia, [s. l.], 29 jun. 2024. Disponível em: https://en.wikipedia.org/wiki/Fail-safe.

Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (1996). *Handbook of Applied Cryptography.* CRC Press.

RSA (sistema criptográfico). Wikipédia: a enciclopédia livre, [s. l.], 28 jun. 2024. Disponível em: https://pt.wikipedia.org/wiki/RSA_(sistema_criptogr%C3%A1fico).

Shamir, A. (1979). *How to share a secret.* Communications of the ACM, 22(11), 612–613. https://doi.org/10.1145/359168.359176

Stallings, W. (2016). *Cryptography and Network Security: Principles and Practice.* 7ª ed., Pearson.

Stinson, D. R. (2005). *Cryptography: Theory and Practice*. 3rd ed., Chapman & Hall/CRC.