THALES PAIVA

30/08/2018

PKP-based Signature Scheme

. Faugère, Koussa, Macario-Rat, Patarin, Perret

07/2018

(1) Permuted Kernel Problem

(2) Identification Schemes (IDS)

(3) IDS com PKP

(4) Fiat-Shamir para obter esquema de assinatura

① Permuted Kernel Problem

DEF (PKP)

Entrada: • p primo $\Rightarrow$ $\mathbb{F}_p$ corpo fi

• Matriz $A \in \mathbb{F}_p^{m \times n}$

• Vetor $V \in \mathbb{F}_p^{n}$.

Desafio:

Encontre uma permutação π sobre $(1, \ldots, n)$ tal que

$$A V_\pi = 0 \text{ , onde}$$

$$V_\pi = (V_{\pi(j)})_{j=1,\ldots,n}$$

EXEMPLO PKP

- Tome $p=5 \Rightarrow F = \mathbb{F}_5$

- $A = \begin{bmatrix} 2 & 1 & 4 & 2 & 0 \\ 4 & 4 & 1 & 0 & 0 \\ 2 & 0 & 4 & 3 & 3 \end{bmatrix}$

$\Rightarrow$ $\text{Ker}(A)$ é gerado por $K = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 2 & 0 \\ 1 & 2 \end{bmatrix}$

$\Rightarrow$ $A.K = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$. Lembrar que estamos em $\mathbb{F}_5 \Rightarrow \text{mod } 5$.

Desafio: Encontre uma permutação dos elementos de $V = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 2 \\ 3 \end{bmatrix}$ que esteja em $\text{Ker}(A)$.

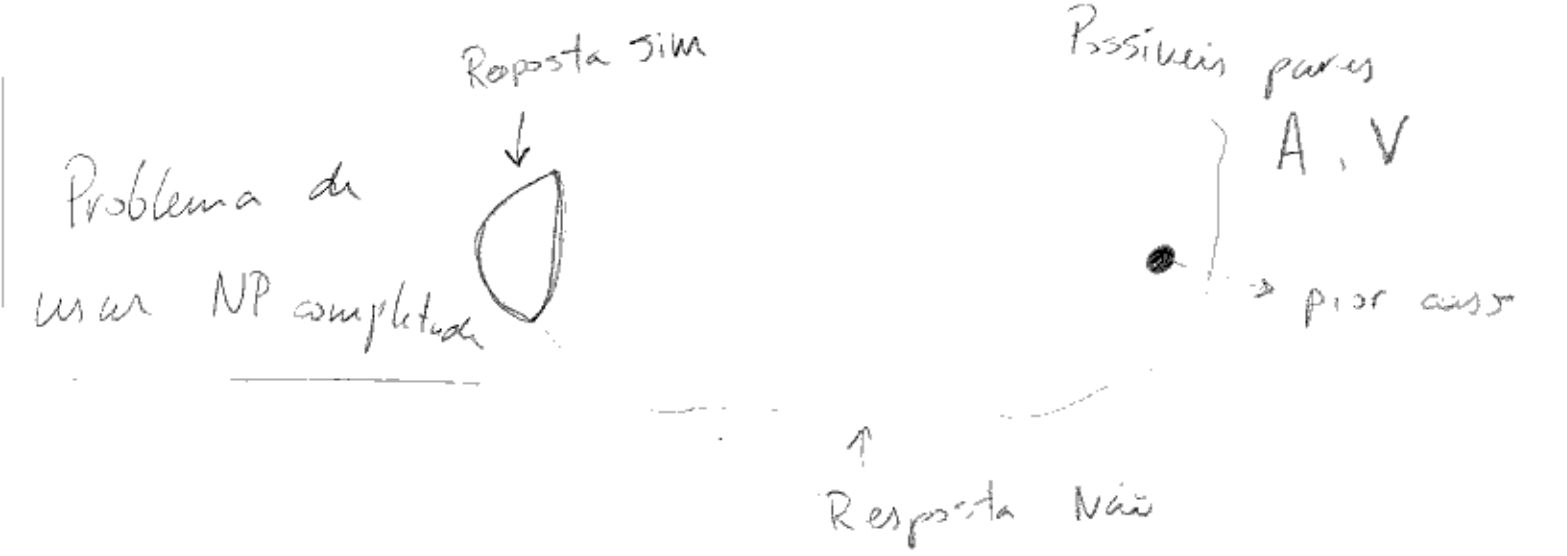
Resposta: $V_\pi = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 2 \\ 0 \end{bmatrix}$

$$\boxed{\pi = (2, 3, 5, 4, 1)}$$

Dificuldade do PKP

- Versão de decisão é NP-completa fortemente ⇒ continua exponencial mesmo se a entrada for codificada em unário.
- Redução: 3SAT → 3DimensionalMatching → 3Partition → PKP

- Nenhum algoritmo conhecido é muito melhor que testar todas as permutações.

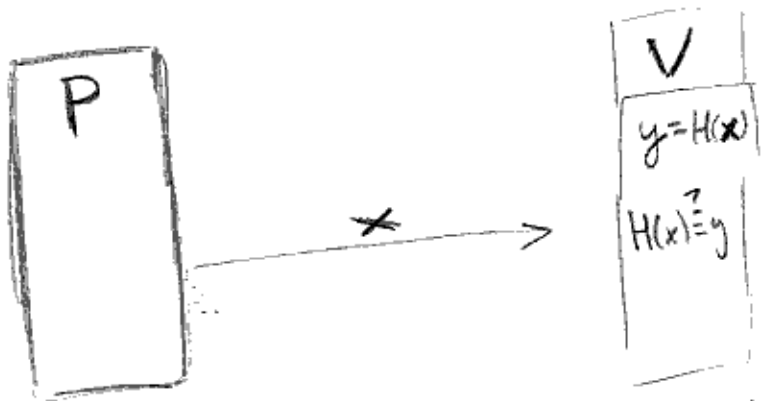


ESQUEMAS DE IDENTIFICAÇÃO

- IDEIA: Alguém (Provador P) quer provar para outro (Verificador V) que P é de fato quem afirma ser.

- Inútil se a comunicação for feita na rede → por si só. Precisa de autenticação.

. SOLUÇÃO SIMPLES (E RUIM) SENHAS



→ (usar exemplo da porta)

Problema: @qualquer um vê a chave passando na rede.

DEF: Um ESQUEMA DE IDENTIFICAÇÃO

Consiste em três algoritmos PPT:

- GEN: Geração de chaves
  Entrada: nível de segurança
  Saída: par (SK, PK)

- Prover: Entrada: (SK, PK)

- Verifier: Entrada: PK.

Prover e Verifier rodam um protocolo

P(A(PK) enganar V(PK)) deve ser desprezível

- Note que a solução SENHA satisfaz def IDS

Noção (básica) importante para IDS

DEF Um IDS (Gen, P, V) é seguro contra um adversário passivo se, para todo Algoritmo PPT $A = (A_1, A_2)$

$$TP(s \leftarrow A_1^{Trans_{SK,PK}}(pk) : \langle A_2(s), V(pk) \rangle = 1)$$

é desprezível.

$\Rightarrow$ Mesmo que um adversário veja várias interações (porém número não mais que polinomial em λ), ele não deve ser capaz de enganar o verificador.

EXEMPLO SCHNORR

$x = SK$ $\qquad$ $y = g^x = PK$

Grupo G é público, com gerador público g de ordem q

$P(x)$		$V(y = g^x)$
$s \leftarrow \mathbb{Z}_q$		
$I := g^s$	$\xrightarrow{I}$	
		$c \leftarrow \mathbb{Z}_q$
	$\xleftarrow{c}$	
$r = cx + s \bmod q$		
	$\xrightarrow{r}$	$g^r \cdot y^{-c} \stackrel{?}{=} I$

① O que um adversário aprende vendo a transcrição?

R. r aleatório.
c aleatório
I calculável a partir de r e c!!

→ Nada

- Por que confiar no esquema?

Suponha que um atacante consegue enganar o verificador com probabilidade não desprezível. Então existem ao menos dois desafios que ele responde corretamente para I. $(c_1 \neq c_2)$

$\Rightarrow$ Transcrições (I, c_1, r_1) e (I, c_2, r_2)

$$\Rightarrow g^{r_1} \cdot y^{-c_1} = g^{r_2} \cdot y^{-c_2}$$

$$\Rightarrow g^{r_1 - r_2} = y^{c_1 - c_2} = g^{x(c_1 - c_2)}$$

$$\Rightarrow r_1 - r_2 = x(c_1 - c_2)$$

Como $c_1 \neq c_2 \Rightarrow c_1 - c_2 \neq 0 \Rightarrow (c_1 - c_2)^{-1}$ existe $\bmod q$

$\Rightarrow$ Log discreto quebrado.

IDS com PKP

- Informações Públicas Globais
  - Número primo p
  - Matriz $m \times n$ $A \in \mathbb{F}_p^{m \times n}$

GEN

- Tome $u \leftarrow \text{Ker}(A)$ aleatório.
  (Lembrar $v \in \text{Ker}(A) \iff Av = 0$)
- Escolha uma permutação π aleatória de S_n. $SK = \pi$
- $v = u_{\pi^{-1}} \Rightarrow Av_\pi = 0$.
- $PK = (v, A, p)$
  ↳ pode usar os λ bits aleatórios p/ representar

Protocolo PKP1DS Shamir '89

Prover		Verifier
$t \leftarrow \mathbb{F}_p^n$		
$\sigma \leftarrow S_n$	$I_0 = H(\sigma, At)$	$c \leftarrow \{0, \dots, p\}$
	$I_1 = H(\sigma\pi, t_\sigma)$	
	$\xleftarrow{c}$	
$w = t_\sigma + c\, v_{\pi\sigma}$		
	$\xrightarrow{w \in \mathbb{F}_p^n}$	
		$b \leftarrow \{0, 1\}$
	$\xleftarrow{b}$	
Se $b = 0$: $\Rightarrow \alpha = \sigma$		
Se $b = 1$: $\Rightarrow \alpha = \pi\sigma$	$\xrightarrow{\alpha}$	Se $b = 0$ $\Rightarrow I_0 \stackrel{?}{=} H(\alpha, A_\alpha w)$
		Se $b = 1$ $H(\alpha, w - c\, v_\alpha) \stackrel{?}{=} I_1$

Corretude! Se Prover é honesto, Verifier aceita.

- $w = t_\sigma + c\, v_{\pi\sigma}$
- Se $b = 0 \Rightarrow \alpha = \sigma$

$\Rightarrow$ $A_\alpha w = A_\sigma (t_\sigma + c\, v_{\pi\sigma})$

$= A_\sigma t_\sigma + A_\sigma c\, v_{\pi\sigma}$

$= At + c \underbrace{A v_\pi}_{0} = At.$

$\Rightarrow H(\alpha, A_\alpha w) = H(\sigma, At) = I_1$ ✓

- Se $b = 1 \Rightarrow \alpha = \pi\sigma$

$\Rightarrow w - c v_\alpha = (t_\sigma + c v_{\pi\sigma}) - c v_{\pi\sigma}$

$= t_\sigma$

$\Rightarrow H(\alpha, w - c v_\alpha) = H(\pi\sigma, t_\sigma) = I_2$

CONHECIMENTO ZERO] Qualquer um pode gerar uma transcição válida com distrib. indistinguível de uma real.

. Fixe um c qualquer

. Fixe b qualquer.

Se $b = 0$:

. Tome $\alpha = \sigma$ aleatório, e t aleatório

Queremos:

$$A_\alpha w = At$$

$$\Rightarrow \text{tome } w = t_\alpha .$$

Se $b = 1$:

. Tome $\alpha = \pi\sigma$ com π, σ aleatórios

Tome t aleatório. Queremos

$$t_\sigma = w - c v_{\pi\sigma}$$

Tome $v_{\pi\sigma}$ aleatório

$$\Rightarrow w = t_\sigma + c v_{\pi\sigma} .$$

$\Rightarrow$ Nenhuma diferença entre esta evic e uma real.

Soundness: A segurança está na disponibilidade do provador, não nas respostas em si.

- Quem sabe os b's pode burlar o sistema
- ϵ's só são importantes para a redução de segurança (acho)

Probabilidade de Fraude

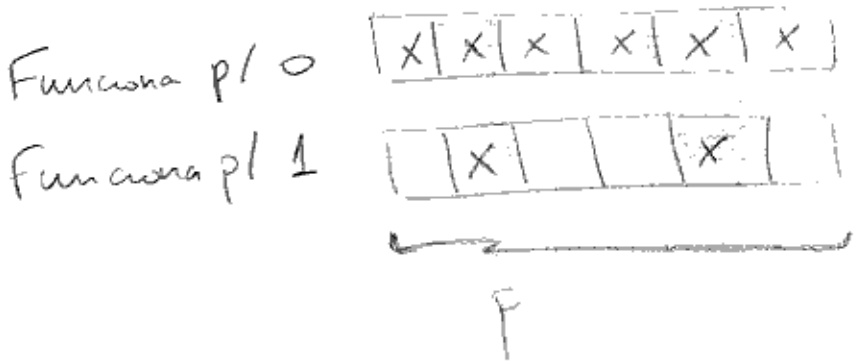
Suponha que um adversário seja capaz de responder $p+2$ desafios corretamente. Então há ao menos dois pares de $\epsilon : c_A, c_B$ tais que

w_A , w_B , funcionam para $b=0$ e $b=1$

$\| \tau = \pi \sigma \|$

valores comitados

Pq? Argumento de PCP



Então

- $I_1 := H(\sigma, At) = H(\sigma, A_\sigma w_A)$

$= H(\sigma, A_\sigma w_B)$

A menos de prob neg de segunda preim

"$\Rightarrow$" $A_\sigma w_A = A_\sigma w_B \Rightarrow$

$$A_\sigma (w_A - w_B) = 0$$

- $I_2 = H(\gamma, t_\sigma) = H(\beta, w_A - c_A v_\gamma)$

$= H(\beta, w_B - c_B v_\gamma)$

"$\Rightarrow$" $w_A - c_A v_\gamma = w_B - c_B v_\gamma$

$$w_A - w_B = (c_A - c_B) v_\gamma$$

$\Rightarrow$

$v_\gamma \in Ker(A)$

$\Rightarrow$ Adversário resolve PKP com perm $\gamma = \Pi\sigma$

Importante Atacante não sabe π!

Conclusão Saber $p+2$ respostas implica quebrar PKP. Note que as respostas não podem ser aprendidas assistindo a interações honestas. Portanto Adversário sabe no máximo $p+1$ respostas.

$$\text{Prob de fraude} = \frac{p+1}{2p} \approx \frac{1}{2}$$

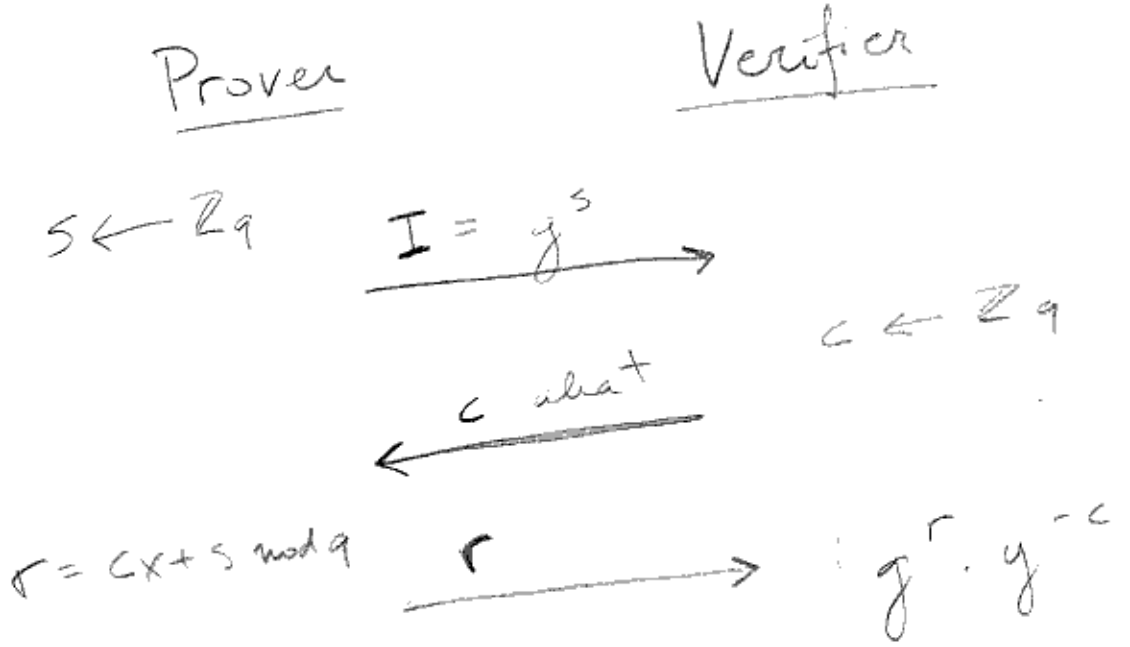
Total de perguntas "|c|.|b|"

Para melhorar $\Rightarrow$ Rodar várias vezes.

Transformação de Fiat-Shamir

IDS → DSS

Tome um IDS como Schnorr



· Um adversário só consegue gerar Transcrições válidas quando sabe c antes de escolher s. (Visto)

· E se forçarmos $c = H(I)$?

⇒ c é sempre escolhido depois de s, e é difícil dar resposta r sem saber sk!

Ideia de Fiat-Shamir:

Fazer $c = H(I, m)$ onde m é a mensagem a ser assinada.

Não só c é aleatório, mas é dependente de m!

- A assinatura é (I, r)

⇒ Somente quem conhece SK pode assinar, pois r é resposta de um IDS.

- Verificação:

calcula $c = H(I, m)$

Aceita se e só se (I, c, r) é interação válida do IDS com PK.

Aplicando Fiat-Shamir ao PKP IDS:

Prover (Verifier do Prover)

$t \leftarrow F_p^n$ $\quad I_0 = H(\sigma, At)$ $\longrightarrow$ $c = H(I_0, I_1, m)$

$\sigma \leftarrow S_n$ $\quad I_1 = H(\sigma\pi, t_\sigma)$

$\longleftarrow c$

$w = t_\sigma + c\, v_{\pi\sigma}$

$\longrightarrow w$

$b = H(c, w, m) \bmod 2$

$\longleftarrow$

Se $b = 0$

$\alpha = \sigma$

Se $b = 1$

$\alpha = \pi\sigma$

Assinatura é (I_0, I_1, w, α)

Porém fazer N rodadas paralelas!

Cada uma o Adv tem $\frac{1}{2}$ chance de fraudar!