**Criptografia - Prova 2 - junho 2020**

Esta prova *não-presencial* é aplicada de acordo com a Resolução da Comissão de Graduação (CoG) da USP, número 7949, de 27 de abril de 2020.

- Horário da prova: das 16h às18h de 15–junho-2020. (Se fosse presencial seria também nesse intervalo.)
- Favor entregar a sua solução *preferencialmente* na forma de texto digitado. Mas ela pode ser manuscrita, contanto que seja bem *legível*; favor verificar esse fato ANTES da entrega.
- Entregar a sua solução *obrigatoriamente* no sistema Paca em forma digital, até, no máximo, às 20h do dia 15-junho-2020. Se houver mais de um arquivo, compactá-los em um *único* arquivo.
- Se for entregue entre 20h e 21h, a nota final terá 1 ponto a menos.

**(Questão 1)-50%-** Esta questão é sobre um algoritmo para Alice se identificar perante um verificador chamado Beto. Esta questão supõe as hipóteses a seguir:

1. Existe uma entidade idônea T que escolhe o primo $p > 3$ tal que $p-1$ seja divisível por um outro primo $q$.
2. T escolhe um elemento $b : 1 \leq b \leq p-1$ tal que a ordem multiplicativa de $b$ seja $q$ (*e.g.,* se $g$ é um gerador $\bmod p$, $b = g^{(p-1)/q} \bmod p$).
3. Cada pessoa como Alice obtém uma cópia autêntica dos parâmetros de T, $cert_T = (p, q, b, T_{pub}, A_T(p, q, b, T_{pub}))$, onde $T_{pub}$ é a chave pública de T e $A_T(p, q, b, T_{pub})$ é a assinatura de T sobre $(p, q, b, T_{pub})$.
4. Um parâmetro $t$ tal que $2^t < q$ é escolhido por T.

## Escolha dos parâmetros para cada usuário

1. Cada pessoa como Alice recebe uma identificação única $I_A$, que contém seus dados pessoais como nome completo, data de nascimento, CPF, etc.
2. Alice escolhe uma chave **secreta** $s$ tal que $1 \leq s \leq q-1$ e calcula a chave **pública** $v = b^{-s} \bmod p$.
3. Alice se identifica perante T por um meio convencional e transfere $v$ para T com integridade, e obtém de T um certificado $cert_A(I_A, v, A_T(I_A, v))$ onde $A_T(I_A, v)$ é a assinatura de T sobre $(I_A, v)$. □

## Protocolo de identificação

Alice se identifica perante um verificador Beto como segue:

1. Alice escolhe um inteiro aleatório $r : 1 \leq r \leq q-1$ e calcula o *testemunho* $x = b^r \bmod p$ e envia para Beto o par $(cert_A, x)$.
2. Beto envia para Alice um inteiro aleatório tal que $1 \leq e \leq 2^t$.
3. Alice verifica se $1 \leq e \leq 2^t$ e envia para Beto (a *resposta*) $y = (s \times e + r) \bmod q$
4. Beto calcula $z = b^y v^e \bmod p$ e aceita a identidade de Alice se, e só se, $z = x$. □

## Esta questão consiste nos seguintes itens:

1. Calcular o comprimento em bits das mensagens $x, e, y, z$
2. Suponha que haja um mal-intencionado Carlos que *não* conhece o segredo $s$ e que tenha armazenado várias mensagens *verdadeiras* trocadas entre a Alice verdadeira e o Beto e queira personificar a Alice. Ele pode ser o próprio verificador Beto. Escrever a *definição* do problema computacional que protege a chave particular e secreta $s$ ou algum outro parâmetro crítico para a segurança. **Só** o nome do problema não basta.
3. Definir o que é protocolo de identificação Zero Knowledge. E justificar o fato deste protocolo ser Zero Knowledge.
4. Descrever e justificar o algoritmo para Beto verificar (à distância) que $p, q, b$ são da autoridade T verdadeira.
5. Descrever e justificar o algoritmo para Beto verificar (à distância) que $v$ é da Alice verdadeira e não de uma outra pessoa.
6. Demonstrar algebricamente que, se Alice de fato conhece $s$, então a igualdade $z = x$ é verdadeira, e *vice-versa*.
7. Descrever algebricamente e demonstrar como um mal-intencionado Carlos, sem conhecer o segredo $s$, poderia personificar Alice (i.e., fazer o Beto aceitar Carlos como sendo a Alice verdadeira). Qual a probabilidade desta personificação obter sucesso?
8. Que personificação (ataque) seria possível se o testemunho $r$ fosse constante ao invés de ser

aleatório?

9. Que personificação (ataque) seria possível se o desafio $e$ fosse constante ao invés de ser aleatório?
10. Quais parâmetros podem ser calculados previamente, antes do protocolo ser executado?

**(Questão 2)-50%-** Esta questão é sobre um algoritmo para Alice assinar um documento digital. Esta questão supõe as hipóteses a seguir:

- É dada uma curva elíptica $E$ sobre $Z_p$ para um primo $p > 3$.
- É dada uma função apropriada e fixa $H()$ de espalhamento (hashing), pública, com probabilidade de ocorrer colisão próxima de zero.
- Nestas condições:

1. O conjunto de legíveis é $Z_p^* \times Z_p^*$
2. O conjunto de ilegíveis é $E \times Z_p^* \times Z_p^*$
3. A chave *pública* é um par de pontos $(Q,P)$ de $E$
4. A chave *secreta* é $s$ tal que $Q = sP$

Algoritmo para Alice assinar uma mensagem $m$

1. Escolher $k \in Z_p$ e calcular $(x_1,y_1) = kP$ e $R = x_1 \bmod p$. Se $R = 0$, repetir este passo até obter $R \neq 0$
2. Calcular $k^{-1} \bmod p$, $A = k^{-1}[H(m) + s \times R] \bmod p$. Se $A = 0$ voltar para o passo (1) anterior até obter $A \neq 0$.
3. A assinatura sobre $m$ é o par $(R,A)$□

Algoritmo para Beto verificar a assinatura

Para verificar a assinatura $(R,A)$ sobre $m$ com a chave pública $Q$

1. Se $R$ e $V$ não satisfizerem $1 \leq R \leq p-1, 1 \leq A \leq p-1$ , rejeitar a assinatura
2. Calcular $A^{-1} \bmod p$, $u_1 = H(m)(A)^{-1} \bmod p$ e $u_2 = R(A)^{-1} \bmod p$
3. Calcular $u_1P + u_2Q = (x_0,y_0)$ e $V = x_0$
4. Aceitar a assinatura se, e somente se, $V = R$. □

Esta questão consiste nos seguintes itens:

1. A assinatura de uma mensagem qualquer, $m$, exige a presença do autor da assinatura? Por quê?
2. A verificação de uma assinatura exige a presença do autor da assinatura? Por quê?
3. Suponha que haja um mal-intencionado Carlos que *não* conhece o segredo $s$ e que tenha armazenado várias assinaturas *verdadeiras* $(R,A)$ e as correspondentes mensagens $m$. Ele pode ser o próprio verificador Beto. Ele deseja falsificar uma assinatura sobre um outro $m_u, m_u \neq m$. Escrever a **definição** do problema computacional que protege a chave particular e secreta $s$ ou algum outro parâmetro crítico para a segurança. **Só** o nome do problema não basta.
4. Dado $(m,R,A)$ demonstrar algebricamente que no *Algoritmo para verificar* acima, se $V = R$, então a assinatura é da Alice verdadeira sobre $m$, e $m$ não foi alterada.
5. E demonstrar a afirmação inversa, ou seja, se a assinatura for da Alice verdadeira sobre $m$ e $m$ não foi alterada, então deve-se ter $V = R$
6. O que ocorre se $m$ for alterada para um outro valor $m_u$ tal que $m_u \neq m$ ? Justifique a sua resposta.
7. Suponha que exista um mal-intencionado Carlos que não conhece o segredo $s$. Mas possui duas assinaturas verdadeiras que foram calculadas com um **mesmo** valor $k$ para duas mensagens distintas $m_1, m_2$. Nestas condições, demonstrar algebricamente que esse Carlos pode falsificar uma assinatura válida para uma terceira mensagem distinta $m_3$. Qual é a probabilidade de sucesso dessa falsificação? A sua resposta deve levar em consideração o comprimento dos parâmetros envolvidos.
8. Suponha que Beto *não* verifica se $R$ e $A$ satisfazem $1 \leq R \leq p-1, 1 \leq A \leq p-1$ no passo (1) da verificação e não rejeita uma assinatura nesse passo. Suponha ainda que esse Carlos (que *não* conhece o segredo $s$ da Alice) possua *uma* assinatura *verdadeira* $(R,A)$ e a correspondente mensagem $m$. Nestas condições, demonstrar algebricamente que Cralos pode gerar uma assinatura falsa, que é aceita, sobre uma mensagem $m_u, m_u \neq m$. Qual é a probabilidade de sucesso de tal falsificação? A sua resposta deve levar em consideração o comprimento dos parâmetros envolvidos.
9. Suponha que esse mal-intencionado Carlos (que *não* conhece o segredo $s$ da Alice) possua várias, $L > 1$, assinaturas *verdadeiras* $(R,A)$ e as correspondentes mensagens $m$. Descrever algebricamente como Carlos poderia calcular o segredo $s$. Qual é a probabilidade de sucesso de cálculo? A sua resposta deve levar em consideração o comprimento dos parâmetros envolvidos.
10. A geração de uma assinatura demora mais ou menos que a verificação da mesma assinatura? Justifique a sua resposta em termos de número de operações básicas relativamente mais demoradas.