# Implementação de protocolos de acordo de chave em dispositivos de poder computacional restrito

Rafael Will Macedo de Araujo
(rwill@ime.usp.br)

DCC – IME – USP

Junho de 2013

CNPq no. 151134/2010-3

# Organização da apresentação

- Conceitos matemáticos importantes
- Acordo de chave
- Modelos alternativos de criptografia de chave pública
  - Baseado em identidade
  - Sem certificado
- Protocolos de acordo de chave nos modelos apresentados
- Experimentos
  - Dispositivos utilizados
  - Custo das operações
  - Custo dos protocolos
  - Uso de memória
  - Troca de mensagens
- Conclusão e sugestões

# Roteiro

1 Introdução

2 Modelos Alternativos

3 Protocolos

4 Ambiente de Testes

5 Resultados

6 Conclusão

# Curvas Elípticas sobre Corpos Finitos

- Seja $\mathbb{F}_q$ um corpo finito de ordem $q$, onde $q > 3$ é um número primo.
  - Como $q$ é primo, então $\mathbb{F}_q$ coincide com $\mathbb{Z}_q$.
- Uma curva elíptica $E$ sobre um corpo $\mathbb{F}_q$, representada por $E(\mathbb{F}_q)$ ou $E/\mathbb{F}_q$, é o conjunto dos pares $(x, y) \in \mathbb{F}_q \times \mathbb{F}_q$ que satisfazem a equação:

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 ,$$

onde $a_1, a_2, a_3, a_4, a_6 \in \mathbb{F}_q$, com $\Delta \neq 0$.

# Curvas Elípticas sobre Corpos Finitos

- $\Delta$ é o discriminante de $E(\mathbb{F}_q)$, e é definido como:

$$\begin{cases} \Delta = -d_2^2 d_8 - 8d_4^3 - 27d_6^2 + 9d_2 d_4 d_6 \\ d_2 = a_1^2 + 4a_2 \\ d_4 = 2a_4 + a_1 a_3 \\ d_6 = a_3^2 + 4a_6 \\ d_8 = a_1^2 a_6 + 4a_2 a_6 - a_1 a_3 a_4 + a_2 a_3^2 - a_4^2 \end{cases}$$

- O discriminante $\Delta \neq 0$ garante que não existirão duas ou mais retas tangentes distintas para um ponto na curva.
- Existe um ponto $\mathcal{O}$, dito ponto no infinito.

# Operação sobre pontos em Curvas Elípticas

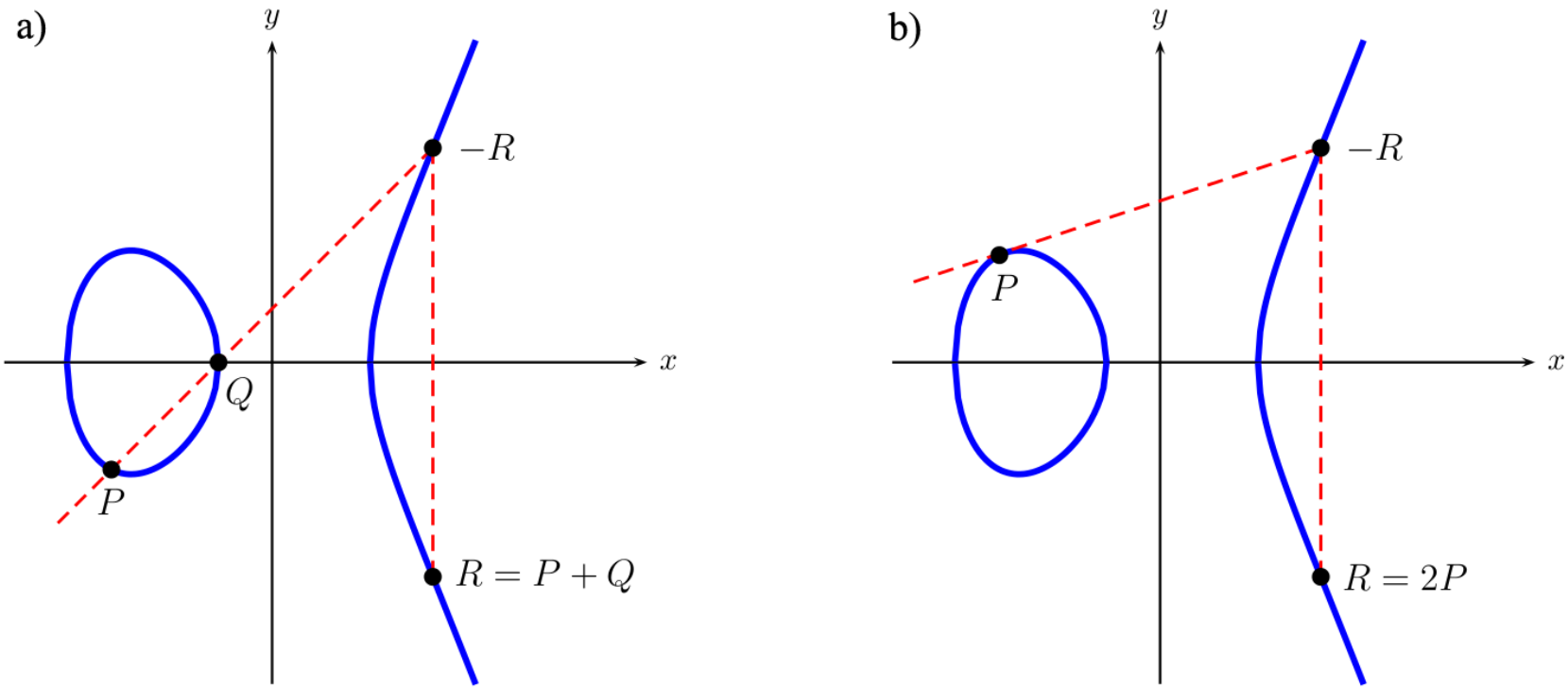



Figura: **a)** soma de dois pontos $P$ e $Q$ e **b)** dobramento de um ponto $P$ em $E(\mathbb{F}_q)$

# Emparelhamento Bilinear

## Definição

- Mapeamento: $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$.

## Propriedades

- **Bilinear**: Para quaisquer $P, Q, R \in \mathbb{G}_1$ ou $\mathbb{G}_2$, e $a, b \in \mathbb{Z}_q$ tem-se:
  - $e(P + Q, R) = e(P, R).e(Q, R)$.
  - $e(P, Q + R) = e(P, Q).e(P, R)$.
  - $e(aQ, bQ) = e(Q, Q)^{ab} = e(abQ, Q) = e(Q, abQ)$.
- **Não-degenerado**: não mapeia todos os pares $\mathbb{G}_1 \times \mathbb{G}_2$ para o elemento identidade de $\mathbb{G}_T$.
- **Ser computável**: $\forall(P, Q), \exists R \rightarrow R \equiv e(P, Q)$.
  - Isto é, existe algoritmo **eficiente** (de tempo polinomial) para calcular $e(P, Q)$.

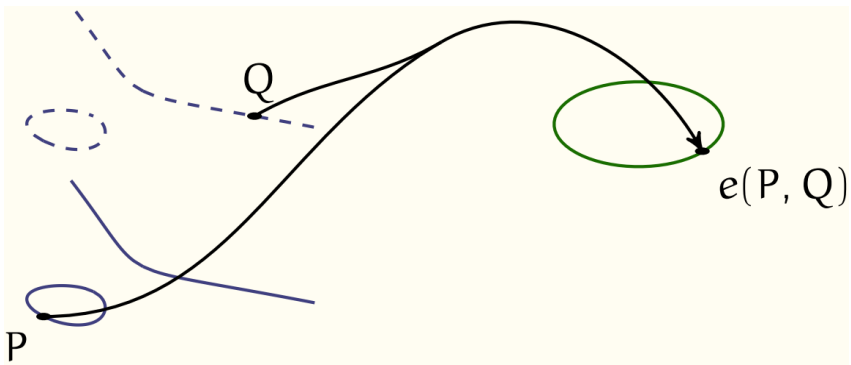



Figura: Emparelhamento bilinear [Cesena, 2010]

# Emparelhamento Bilinear

## Definição

- Mapeamento: $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$.

## Propriedades

- **Bilinear**: Para quaisquer $P, Q, R \in \mathbb{G}_1$ ou $\mathbb{G}_2$, e $a, b \in \mathbb{Z}_q$ tem-se:
  - $e(P + Q, R) = e(P, R).e(Q, R)$.
  - $e(P, Q + R) = e(P, Q).e(P, R)$.
  - $e(aQ, bQ) = e(Q, Q)^{ab} = e(abQ, Q) = e(Q, abQ)$.
- **Não-degenerado**: não mapeia todos os pares $\mathbb{G}_1 \times \mathbb{G}_2$ para o elemento identidade de $\mathbb{G}_T$.
- **Ser computável**: $\forall(P, Q), \exists R \rightarrow R \equiv e(P, Q)$.
  - Isto é, existe algoritmo **eficiente** (de tempo polinomial) para calcular $e(P, Q)$.

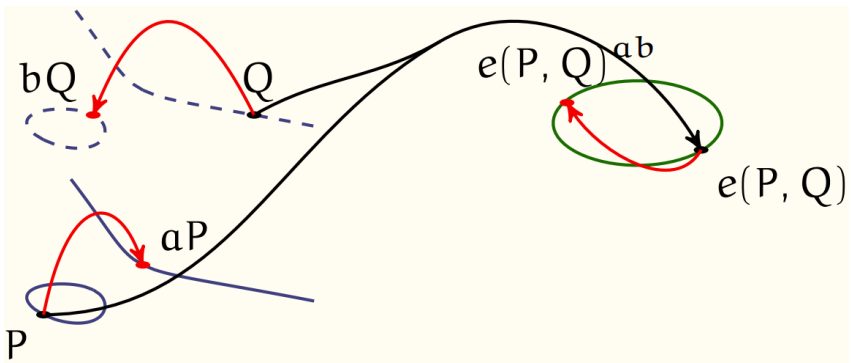



Figura: Emparelhamento bilinear [Cesena, 2010]

# Problemas Computacionais

## Problema do logaritmo discreto

Dados $P$ e $R \in E(\mathbb{F}_q)$, $P$ gerador de $E(\mathbb{F}_q)$, encontrar $s \in \mathbb{Z}_q$ tal que:

$$R = sP$$

OBS: Note que $sP = \underbrace{P + P + \cdots + P}_{s \text{ vezes}}$.

## Problemas *Diffie-Hellman* (DHP)

Considere $P \in \mathbb{G}$ (aditivo) e $a, b, c \in \mathbb{Z}_q$:

- **DDHP** (*Decision*-DHP): dados: $P, aP, bP, cP \in \mathbb{G}$; decidir: $abP = cP$.
- **CDHP** (*Computacional*-DHP): dados: $P, aP, bP \in \mathbb{G}$; encontrar: $abP$.
- **GDHP** (*Gap*-DHP): dados: $P, aP, bP \in \mathbb{G}$; encontrar: $abP$, com ajuda de um oráculo de decisão (que dados $aP, bP, cP \in \mathbb{G}$, decide se $abP = cP$).
- **BDHP** (*Bilinear*-DHP): dados: $P, aP, bP, cP \in \mathbb{G}$; encontrar: $e(P, P)^{abc}$.

# Comprimento Mínimo das Chaves

| Segurança (bits) | Algoritmo simétrico | RSA | ECC | Razão de crescimento |
|---|---|---|---|---|
| 80 | Skipjack | 1024 | 160 | $\approx 1:6$ |
| 112 | 3DES | 2048 | 224 | $\approx 1:9$ |
| 128 | AES-128 | 3072 | 256 | $1:12$ |
| 192 | AES-192 | 7680 | 384 | $1:20$ |
| 256 | AES-256 | 15360 | 512 | $1:30$ |

**Conclusão**: Através do uso de curvas elípticas (ECC - *Elliptic Curve Cryptography*) é possível trabalhar com chaves menores, mantendo um elevado nível de segurança.

# Acordo de Chave

- É o processo no qual dois ou mais participantes combinam uma chave secreta, de modo que ambos influenciam no resultado.

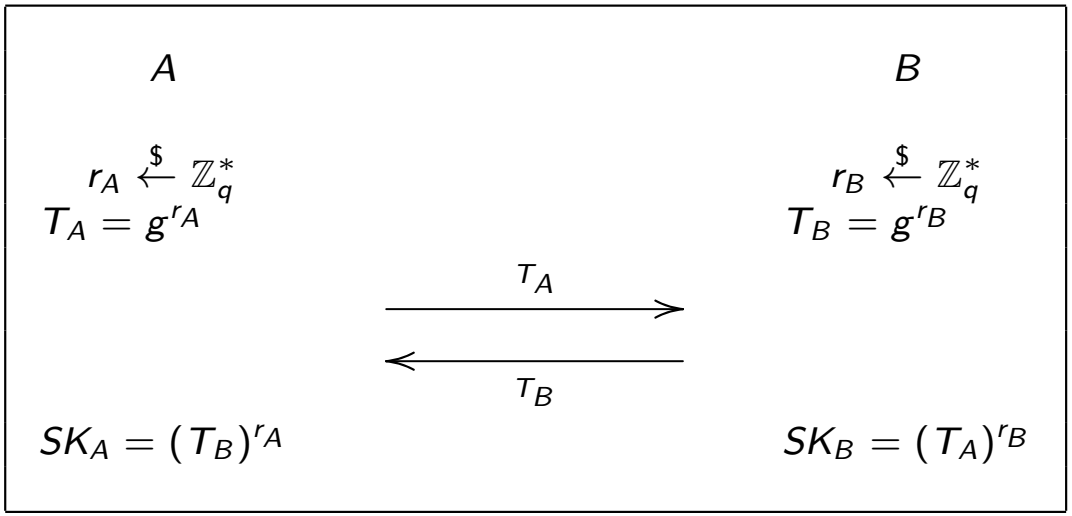



Figura: Acordo de chave Diffie-Hellman (1976)

$$SK_A = (T_B)^{r_A} = (g^{r_B})^{r_A} = (g^{r_A})^{r_B} = (T_A)^{r_B} = SK_B$$

Sendo $g$ um gerador de $\mathbb{Z}_q$ e $q$ um número primo (grande).

# Roteiro

1 Introdução

2 Modelos Alternativos

3 Protocolos

4 Ambiente de Testes

5 Resultados

6 Conclusão

# Modelo Baseado em Identidade (*ID-Based*)

- Modelo proposto em 1984 por A. Shamir.
  - Somente em 2001 foram publicados esquemas de criptografia (*encryption*) no modelo baseado em identidades [Boneh & Franklin, 2001] e [Cocks, 2001].
- A chave pública deixa de ser um valor gerado aleatoriamente, e passa ser a própria **identidade do usuário**.
- Pressupõe a existência de um **Gerador de Chaves Privadas** (PKG - *Private Key Generator*). Trata-se de uma **Autoridade de Confiança**, responsável por:
  - Gerar e guardar a chave-mestra secreta do sistema.
  - Calcular as chaves secretas de todos os usuários.
  - Entregar as chaves secretas dos usuários de forma segura (com sigilo e autenticidade).

# Modelo Baseado em Identidade (*ID-Based*)

Sejam $s \xleftarrow{\$} \mathbb{Z}_q^*$, $P$ um gerador de $E(\mathbb{F}_q)$, $sP \in E(\mathbb{F}_q)$ e seja a função de *hash* $\mathcal{H} : \{0,1\}^* \rightarrow E(\mathbb{F}_q)$.

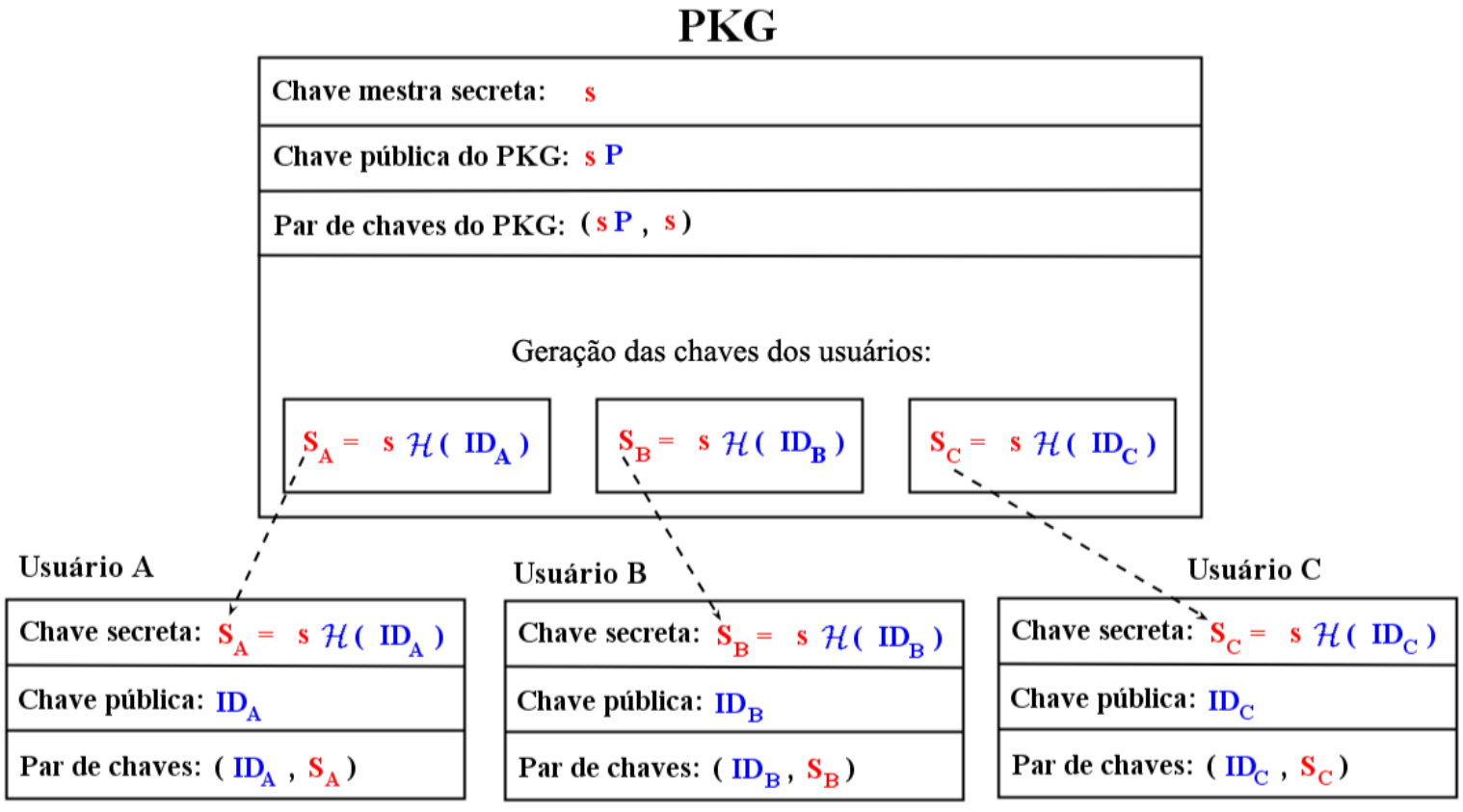



Figura: Diagrama do modelo baseado em identidade

# Modelo Baseado em Identidade (*ID-Based*)

## Vantagens

- Chave pública de fácil memorização.
- Certificação implícita.
- Infraestrutura simples.

## Desvantagens

- Alta criticidade da chave-mestra secreta.
- Custódia de chaves.
- Não há irretratabilidade (o PKG pode personificar qualquer usuário).

# Modelo sem Certificado (*Certificateless*)

- Modelo proposto em 2003 por S. Al-Riyami e K. Paterson.
  - Modelo intermediário entre o baseado em identidade e a ICP.
- Elimina a propriedade de custódia de chaves, inerente ao modelo baseado em identidade.
- Pressupõe a existência de um **Centro Gerador de Chaves** (KGC - *Key Generator Center*), responsável por:
  - Gerar e guardar a chave-mestra secreta do sistema.
  - Calcular as chaves secretas *parciais* de todos os usuários.
  - Entregar as chaves secretas parciais dos usuários de forma segura (com sigilo e autenticidade).

# Modelo sem Certificado (*Certificateless*)

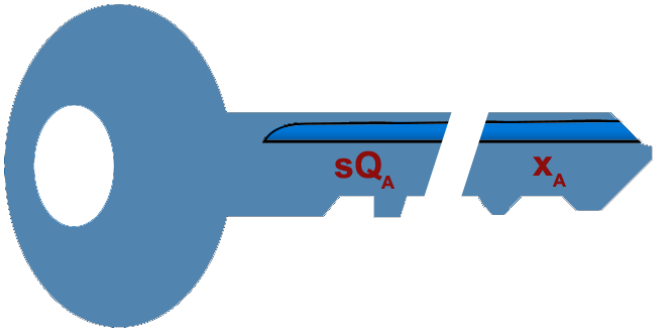



Figura: Chaves secretas parciais

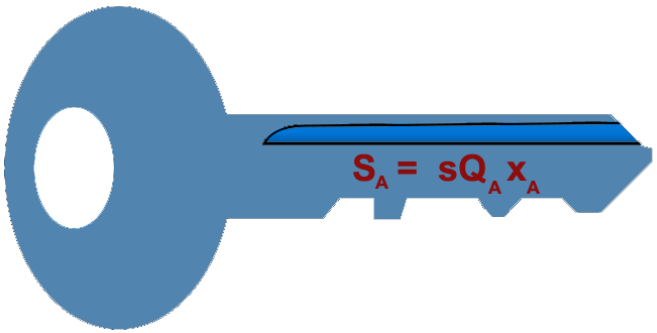



Figura: Chave secreta completa

# Modelo sem Certificado (*Certificateless*)

Sejam $s \xleftarrow{\$} \mathbb{Z}_q^*$, $P$ um gerador de $E(\mathbb{F}_q)$, $sP \in E(\mathbb{F}_q)$ e seja a função de *hash* $\mathcal{H} : \{0,1\}^* \rightarrow E(\mathbb{F}_q)$.

Seja $x_U \xleftarrow{\$} \mathbb{Z}_q^*$ um valor escolhido pelo participante $U$.

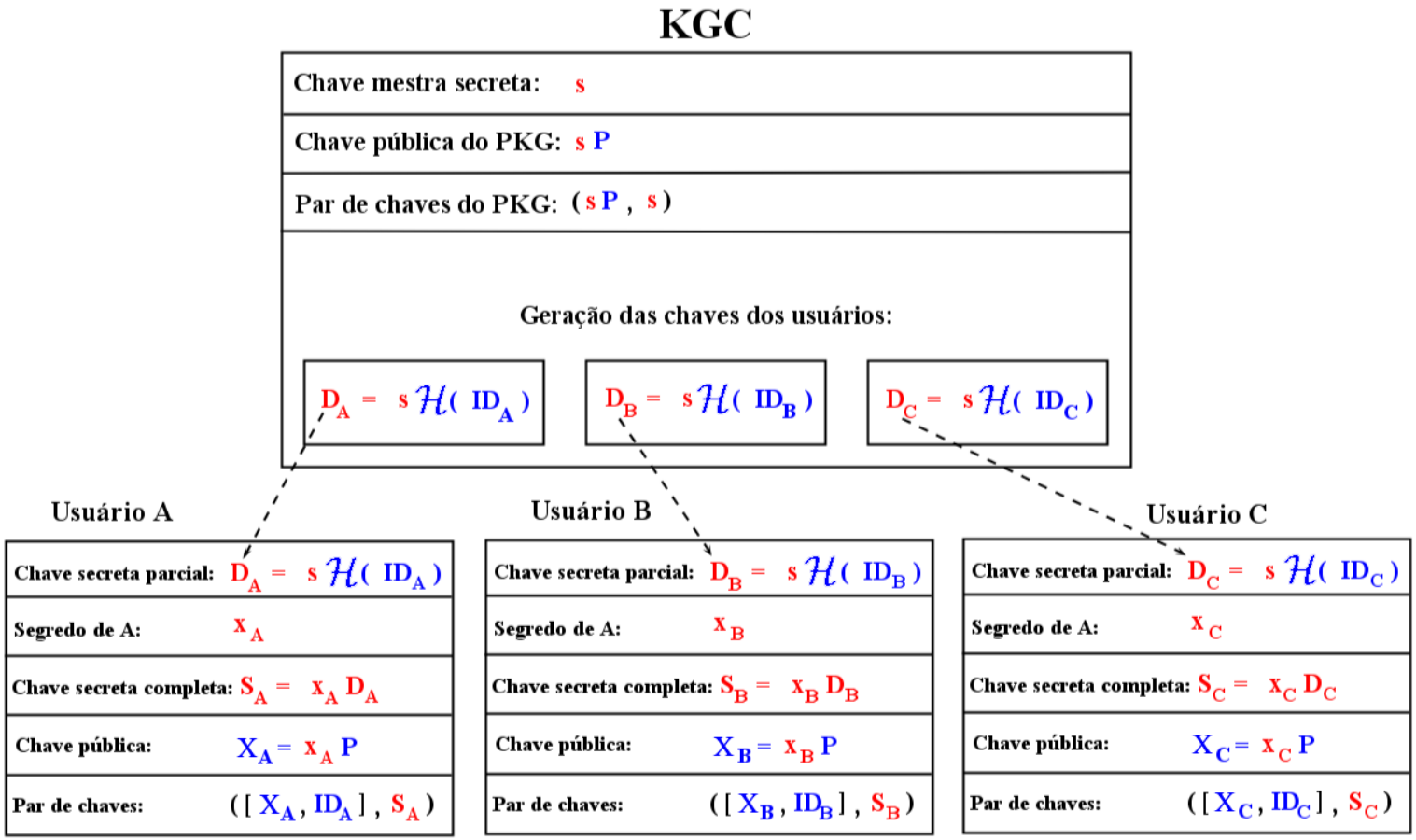



Figura: Diagrama do modelo sem certificado

# Modelo sem Certificado (*Certificateless*)

## Vantagens

- Elimina a custódia de chaves.
- Menor criticidade da chave mestra.
- Controle do usuário sobre a renovação de chaves.
- Certificação implícita.

## Desvantagens

- Requer um repositório ou alguma outra forma de distribuição de chaves públicas.
- Possibilidade de substituição da chave pública por algum mal intencionado (dificuldade em verificar a legitimidade da chave pública).
- Vulnerabilidade ao ataque DoD (*Denial of Decryption*). A substituição de uma chave pública impede que a mensagem seja decifrada.

# Roteiro

1 Introdução

2 Modelos Alternativos

3 Protocolos

4 Ambiente de Testes

5 Resultados

6 Conclusão

# Protocolos Estudados

- **ID-Based**:
  - **HC-BDH**: Huang, Cao. 2009
  - **HLZ-GBDH**: Hu, Liu, Zhang. 2009
  - **CC-BDH**: Chow, Choo. 2007
  - **NCLH-BDH** e NCLH-GBDH: Ni, Chen, Li, Hao. 2011
  - **NCL-BDH**: Ni, Chen, Li. 2012

- **Certificateless**:
  - **LBG-BDH** e **LBG-GBDH**: Lippold, Boyd, Gonzalez Nieto. 2009
  - **GOT-BDH** e **GOT-GBDH**: Goya, Okida, Terada. 2010
  - **GNT1-GBDH**: Goya, Nakamura, Terada. 2011
  - **GNT3-BDH**: Goya, Nakamura, Terada. 2011
  - **GNT2-GBDH**: Goya, Nakamura, Terada. 2011
  - **GNT4-BDH**: Goya, Nakamura, Terada. 2011

# Exemplo 1: Protocolo Huang-Cao, 2009

O PKG publica os seguintes parâmetros: $\langle k, \mathbb{G}, \mathbb{G}_T, q, e, P, sP, \mathcal{H}_0, \mathcal{H}_1, \mathcal{H}_2 \rangle$, onde:

- $k$ é um parâmetro de segurança
- $q$ é a ordem dos grupos $\mathbb{G}$ (aditivo) e $\mathbb{G}_T$ (multiplicativo)
- $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, é um emparelhamento bilinear admissível
- $P$ é um ponto gerador de $\mathbb{G}$
- $sP$ é a chave pública do PKG
- $\mathcal{H}_0, \mathcal{H}_1, \mathcal{H}_2$ são funções de hash tais que: $\mathcal{H}_0 : \{0,1\}^* \to \{0,1\}^k$ e $\mathcal{H}_1, \mathcal{H}_2 : \{0,1\}^* \to \mathbb{G}$

| $A$ | | $B$ |
|---|---|---|
| $Q_{B_1} = \mathcal{H}_1(ID_B)$<br>$Q_{B_2} = \mathcal{H}_2(ID_B)$ | | $Q_{A_1} = \mathcal{H}_1(ID_A)$<br>$Q_{A_2} = \mathcal{H}_2(ID_A)$ |
| $r_A \xleftarrow{\$} \mathbb{Z}_q^*$ | | $r_B \xleftarrow{\$} \mathbb{Z}_q^*$ |
| $R_A = r_A P$ | | $R_B = r_B P$ |
| | $\xrightarrow{R_A}$ | |
| | $\xleftarrow{R_B}$ | |
| $Z_{A_1} = e(R_B + Q_{B_1}, r_A sP + S_{A_1})$<br>$Z_{A_2} = e(R_B + Q_{B_2}, r_A sP + S_{A_2})$<br>$Z_{A_3} = r_A R_B$<br>$sid_A = (R_A, R_B, ID_A, ID_B)$<br>$SK_A = \mathcal{H}_0(Z_{A_1}, Z_{A_2}, Z_{A_3}, sid_A)$ | | $Z_{B_1} = e(R_A + Q_{A_1}, r_B sP + S_{B_1})$<br>$Z_{B_2} = e(R_A + Q_{A_2}, r_B sP + S_{B_2})$<br>$Z_{B_3} = r_B R_A$<br>$sid_B = (R_A, R_B, ID_A, ID_B)$<br>$SK_B = \mathcal{H}_0(Z_{B_1}, Z_{B_2}, Z_{B_3}, sid_B)$ |

# Exemplo 2: Protocolo Goya-Okida-Terada (BDH), 2010

O KGC publica os parâmetros: $\langle k, \mathbb{G}, \mathbb{G}_T, q, e, P, sP, \mathcal{H}_0, \mathcal{H}_1, \mathcal{H}_2 \rangle$.

| $A$ | | $B$ |
|---|---|---|
| $Q_{B_1} = \mathcal{H}_1(ID_B)$<br>$Q_{B_2} = \mathcal{H}_2(ID_B)$<br>$r_A \xleftarrow{\$} \mathbb{Z}_q^*$<br>$R_A = r_A P$<br>$X_A = x_A P$ | | $Q_{A_1} = \mathcal{H}_1(ID_A)$<br>$Q_{A_2} = \mathcal{H}_2(ID_A)$<br>$r_B \xleftarrow{\$} \mathbb{Z}_q^*$<br>$R_B = r_B P$<br>$X_B = x_B P$ |
| | $\xrightarrow{R_A, X_A}$ | |
| | $\xleftarrow{R_B, X_B}$ | |
| $K_{A_1} = e(R_B + Q_{B_1}, r_A sP + D_{A_1})$<br>$K_{A_2} = e(R_B + Q_{B_2}, r_A sP + D_{A_2})$<br>$L_{A_1} = e(Q_{B_1}, sP)^{x_A}.e(D_{A_1}, X_B)$<br>$L_{A_2} = e(Q_{B_2}, sP)^{x_A}.e(D_{A_2}, X_B)$<br>$N_{A_1} = e(Q_{B_1}, D_{A_1})$<br>$N_{A_2} = e(Q_{B_2}, D_{A_2})$<br>$Z_A = (x_A X_B, x_A R_B, r_A R_B, r_A X_B)$<br>$sid_A = (ID_A, ID_B, R_A, R_B, X_A, X_B)$<br>$SK_A = \mathcal{H}_0(sid_A, Z_A, K_{A_1}, K_{A_2}, L_{A_1}, L_{A_2}, N_{A_1}, N_{A_2})$ | | $K_{B_1} = e(R_A + Q_{A_1}, r_B sP + D_{B_1})$<br>$K_{B_2} = e(R_A + Q_{A_2}, r_B sP + D_{B_2})$<br>$L_{B_1} = e(Q_{A_1}, sP)^{x_B}.e(D_{B_1}, X_A)$<br>$L_{B_2} = e(Q_{A_2}, sP)^{x_B}.e(D_{B_2}, X_A)$<br>$N_{B_1} = e(Q_{A_1}, D_{B_1})$<br>$N_{B_2} = e(Q_{A_2}, D_{B_2})$<br>$Z_B = (x_B X_A, r_B X_A, r_B R_A, x_B R_A)$<br>$sid_B = (ID_A, ID_B, R_A, R_B, X_A, X_B)$<br>$SK_B = \mathcal{H}_0(sid_B, Z_B, K_{B_1}, K_{B_2}, L_{B_1}, L_{B_2}, N_{B_1}, N_{B_2})$ |

# Exemplo 2: Protocolo Goya-Okida-Terada (BDH), 2010

Substituição das operações de exponenciação em $\mathbb{G}_T$ por multiplicação em $\mathbb{G}$.

| $A$ | | $B$ |
|---|---|---|
| $Q_{B_1} = \mathcal{H}_1(ID_B)$<br>$Q_{B_2} = \mathcal{H}_2(ID_B)$<br>$r_A \xleftarrow{\$} \mathbb{Z}_q^*$<br>$R_A = r_A P$<br>$X_A = x_A P$ | | $Q_{A_1} = \mathcal{H}_1(ID_A)$<br>$Q_{A_2} = \mathcal{H}_2(ID_A)$<br>$r_B \xleftarrow{\$} \mathbb{Z}_q^*$<br>$R_B = r_B P$<br>$X_B = x_B P$ |
| | $\xrightarrow{R_A, X_A}$ | |
| | $\xleftarrow{R_B, X_B}$ | |
| $K_{A_1} = e(R_B + Q_{B_1}, r_A sP + D_{A_1})$<br>$K_{A_2} = e(R_B + Q_{B_2}, r_A sP + D_{A_2})$<br>$L_{A_1} = e(Q_{B_1}, x_A sP).e(D_{A_1}, X_B)$<br>$L_{A_2} = e(Q_{B_2}, x_A sP).e(D_{A_2}, X_B)$<br>$N_{A_1} = e(Q_{B_1}, D_{A_1})$<br>$N_{A_2} = e(Q_{B_2}, D_{A_2})$<br>$Z_A = (x_A X_B, x_A R_B, r_A R_B, r_A X_B)$<br>$sid_A = (ID_A, ID_B, R_A, R_B, X_A, X_B)$<br>$SK_A = \mathcal{H}_0(sid_A, Z_A, K_{A_1}, K_{A_2}, L_{A_1}, L_{A_2}, N_{A_1}, N_{A_2})$ | | $K_{B_1} = e(R_A + Q_{A_1}, r_B sP + D_{B_1})$<br>$K_{B_2} = e(R_A + Q_{A_2}, r_B sP + D_{B_2})$<br>$L_{B_1} = e(Q_{A_1}, x_B sP).e(D_{B_1}, X_A)$<br>$L_{B_2} = e(Q_{A_2}, x_B sP).e(D_{B_2}, X_A)$<br>$N_{B_1} = e(Q_{A_1}, D_{B_1})$<br>$N_{B_2} = e(Q_{A_2}, D_{B_2})$<br>$Z_B = (x_B X_A, r_B X_A, r_B R_A, x_B R_A)$<br>$sid_B = (ID_A, ID_B, R_A, R_B, X_A, X_B)$<br>$SK_B = \mathcal{H}_0(sid_B, Z_B, K_{B_1}, K_{B_2}, L_{B_1}, L_{B_2}, N_{B_1}, N_{B_2})$ |

# Exemplo 2: Protocolo Goya-Okida-Terada (BDH), 2010

Variáveis que dependem das chaves de sessão.

| $A$ | | $B$ |
|---|---|---|
| $Q_{B_1} = \mathcal{H}_1(ID_B)$<br>$Q_{B_2} = \mathcal{H}_2(ID_B)$<br>$r_A \xleftarrow{\$} \mathbb{Z}_q^*$<br>$R_A = r_A P$<br>$X_A = x_A P$ | | $Q_{A_1} = \mathcal{H}_1(ID_A)$<br>$Q_{A_2} = \mathcal{H}_2(ID_A)$<br>$r_B \xleftarrow{\$} \mathbb{Z}_q^*$<br>$R_B = r_B P$<br>$X_B = x_B P$ |
| | $\xrightarrow{R_A, X_A}$ | |
| | $\xleftarrow{R_B, X_B}$ | |
| $K_{A_1} = e(R_B + Q_{B_1}, r_A sP + D_{A_1})$<br>$K_{A_2} = e(R_B + Q_{B_2}, r_A sP + D_{A_2})$<br>$L_{A_1} = e(Q_{B_1}, x_A sP).e(D_{A_1}, X_B)$<br>$L_{A_2} = e(Q_{B_2}, x_A sP).e(D_{A_2}, X_B)$<br>$N_{A_1} = e(Q_{B_1}, D_{A_1})$<br>$N_{A_2} = e(Q_{B_2}, D_{A_2})$<br>$Z_A = (x_A X_B, x_A R_B, r_A R_B, r_A X_B)$<br>$sid_A = (ID_A, ID_B, R_A, R_B, X_A, X_B)$<br>$SK_A = \mathcal{H}_0(sid_A, Z_A, K_{A_1}, K_{A_2}, L_{A_1}, L_{A_2}, N_{A_1}, N_{A_2})$ | | $K_{B_1} = e(R_A + Q_{A_1}, r_B sP + D_{B_1})$<br>$K_{B_2} = e(R_A + Q_{A_2}, r_B sP + D_{B_2})$<br>$L_{B_1} = e(Q_{A_1}, x_B sP).e(D_{B_1}, X_A)$<br>$L_{B_2} = e(Q_{A_2}, x_B sP).e(D_{B_2}, X_A)$<br>$N_{B_1} = e(Q_{A_1}, D_{B_1})$<br>$N_{B_2} = e(Q_{A_2}, D_{B_2})$<br>$Z_B = (x_B X_A, r_B X_A, r_B R_A, x_B R_A)$<br>$sid_B = (ID_A, ID_B, R_A, R_B, X_A, X_B)$<br>$SK_B = \mathcal{H}_0(sid_B, Z_B, K_{B_1}, K_{B_2}, L_{B_1}, L_{B_2}, N_{B_1}, N_{B_2})$ |

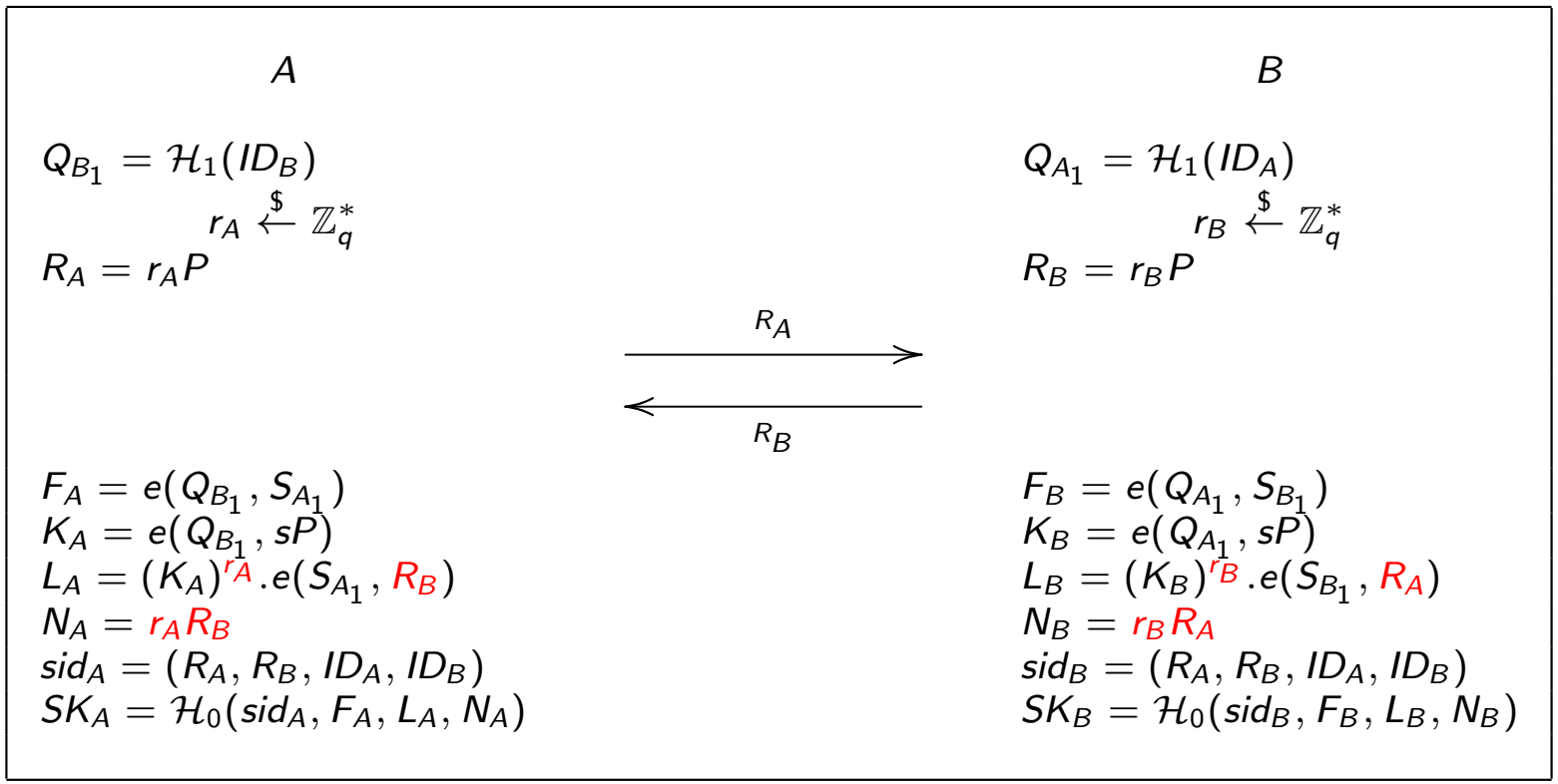
# Exemplo 2: Protocolo Goya-Okida-Terada (BDH), 2010

Caso com pré-computação: pode ser aplicado em comunicações subsequentes.

| $A$ | | $B$ |
|---|---|---|
| $Q_{B_1} = \mathcal{H}_1(ID_B)$ | | $Q_{A_1} = \mathcal{H}_1(ID_A)$ |
| $Q_{B_2} = \mathcal{H}_2(ID_B)$ | | $Q_{A_2} = \mathcal{H}_2(ID_A)$ |
| $r_A \xleftarrow{\$} \mathbb{Z}_q^*$ | | $r_B \xleftarrow{\$} \mathbb{Z}_q^*$ |
| $R_A = r_A P$ | | $R_B = r_B P$ |
| $X_A = x_A P$ | | $X_B = x_B P$ |
| | $\xrightarrow{R_A, X_A}$ | |
| | $\xleftarrow{R_B, X_B}$ | |
| $K_{A_1} = e(R_B + Q_{B_1}, r_A sP + D_{A_1})$ | | $K_{B_1} = e(R_A + Q_{A_1}, r_B sP + D_{B_1})$ |
| $K_{A_2} = e(R_B + Q_{B_2}, r_A sP + D_{A_2})$ | | $K_{B_2} = e(R_A + Q_{A_2}, r_B sP + D_{B_2})$ |
| $L_{A_1}$ | | $L_{B_1}$ |
| $L_{A_2}$ | | $L_{B_2}$ |
| $N_{A_1}$ | | $N_{B_1}$ |
| $N_{A_2}$ | | $N_{B_2}$ |
| $Z_A = ( \quad x_A R_B, r_A R_B, r_A X_B)$ | | $Z_B = ( \quad r_B X_A, r_B R_A, x_B R_A)$ |
| $sid_A = (ID_A, ID_B, R_A, R_B, X_A, X_B)$ | | $sid_B = (ID_A, ID_B, R_A, R_B, X_A, X_B)$ |
| $SK_A = \mathcal{H}_0(sid_A, Z_A, K_{A_1}, K_{A_2}, L_{A_1}, L_{A_2}, N_{A_1}, N_{A_2})$ | | $SK_B = \mathcal{H}_0(sid_B, Z_B, K_{B_1}, K_{B_2}, L_{B_1}, L_{B_2}, N_{B_1}, N_{B_2})$ |

# Exemplo 3: Protocolo Ni-Chen-Li-Hao (GBDH), 2011

Nem sempre compensa trocar uma operação de exponenciação em $\mathbb{G}_T$ por uma multiplicação em $\mathbb{G}$.



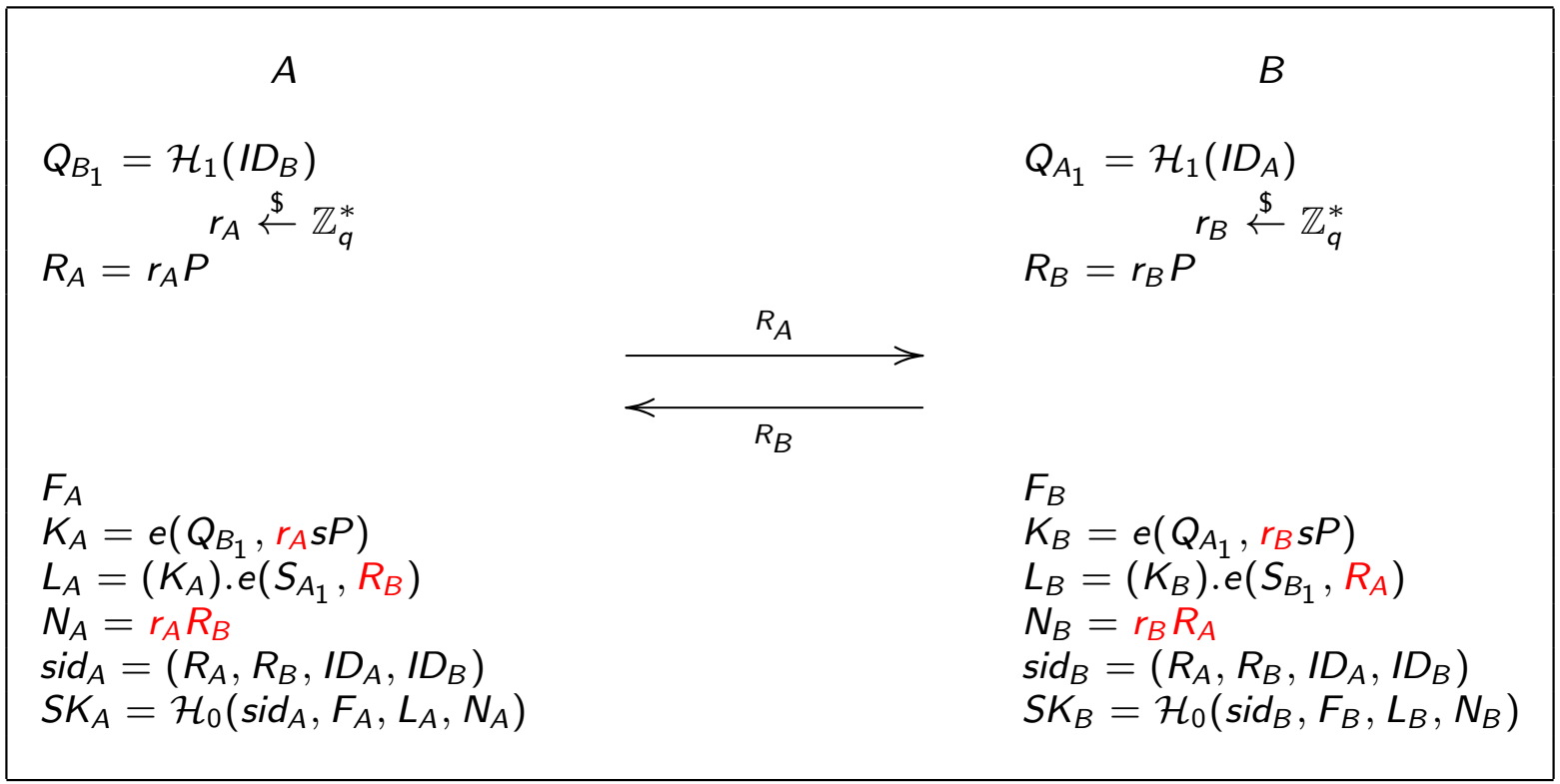
# Exemplo 3: Protocolo Ni-Chen-Li-Hao (GBDH), 2011

Nem sempre compensa trocar uma operação de exponenciação em $\mathbb{G}_T$ por uma multiplicação em $\mathbb{G}$.

| $A$ | | $B$ |
|---|---|---|
| $Q_{B_1} = \mathcal{H}_1(ID_B)$ | | $Q_{A_1} = \mathcal{H}_1(ID_A)$ |
| $r_A \xleftarrow{\$} \mathbb{Z}_q^*$ | | $r_B \xleftarrow{\$} \mathbb{Z}_q^*$ |
| $R_A = r_A P$ | | $R_B = r_B P$ |
| | $\xrightarrow{R_A}$ | |
| | $\xleftarrow{R_B}$ | |
| $F_A$ | | $F_B$ |
| $K_A$ | | $K_B$ |
| $L_A = (K_A)^{r_A}.e(S_{A_1}, R_B)$ | | $L_B = (K_B)^{r_B}.e(S_{B_1}, R_A)$ |
| $N_A = r_A R_B$ | | $N_B = r_B R_A$ |
| $sid_A = (R_A, R_B, ID_A, ID_B)$ | | $sid_B = (R_A, R_B, ID_A, ID_B)$ |
| $SK_A = \mathcal{H}_0(sid_A, F_A, L_A, N_A)$ | | $SK_B = \mathcal{H}_0(sid_B, F_B, L_B, N_B)$ |

# Exemplo 3: Protocolo Ni-Chen-Li-Hao (GBDH), 2011

Eliminar a exponenciação em $\mathbb{G}_T$ gerou 1 emparelhamento a mais, além da multiplicação em $\mathbb{G}$.



# Roteiro

1 Introdução

2 Modelos Alternativos

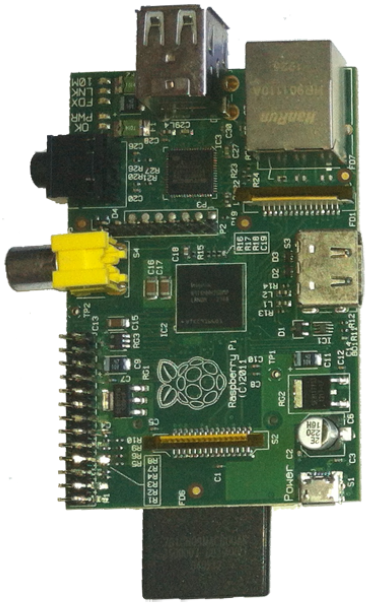
3 Protocolos

4 Ambiente de Testes

5 Resultados

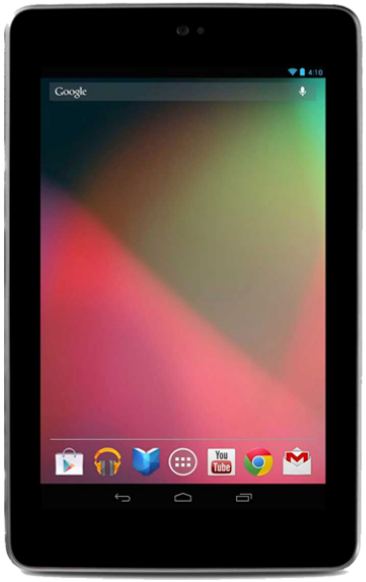
6 Conclusão

# *Hardware* utilizado



Raspberry Pi


Motorola Milestone 1




Asus Google Nexus 7

# *Hardware* utilizado

- **Servidor (PC)**:
  - Processador: Intel Core 2 Duo T5800, 2.0 GHz
  - Memória RAM: 3 GB, DDR2, 800 MHz
  - Conexão de rede: 100 Mbps, por cabo
  - Sistema operacional: Ubuntu 12.04 LTS, 32 bits
- **Raspberry Pi (RPi)**:
  - Processador: Broadcom BCM2835 (intruções ARMv6, *single core*), 700 MHz
  - Memória RAM: 256 MB, LPDDR2
  - Conexão de rede: 100 Mbps, por cabo
  - Sistema operacional: Debian "Wheezy" (armhf)
- **Smartphone (MM1)**:
  - Modelo: Motorola Milestone 1
  - Processador: Cortex-A8 (intruções ARMv7, *single core*), 600 MHz
  - Memória RAM: 256 MB, LPDDR
  - Conexão de rede: 54 Mbps, *Wi-Fi* (IEEE 802.11g)
  - Sistema operacional: Android 2.2
- **Tablet (GN7)**:
  - Modelo: Asus Google Nexus 7
  - Processador: Cortex-A9 (intruções ARMv7, *quad-core*), 1.2 GHz
  - Memória RAM: 1 GB, DDR3
  - Conexão de rede: 54 Mbps, *Wi-Fi* (IEEE 802.11g)
  - Sistema operacional: Android 4.2
- **Roteador Wi-Fi**:
  - Modelo: D-Link DIR-300
  - Conexão de rede: 4x 100 Mbps (LAN), 54 Mbps (WLAN)

# Comunicação

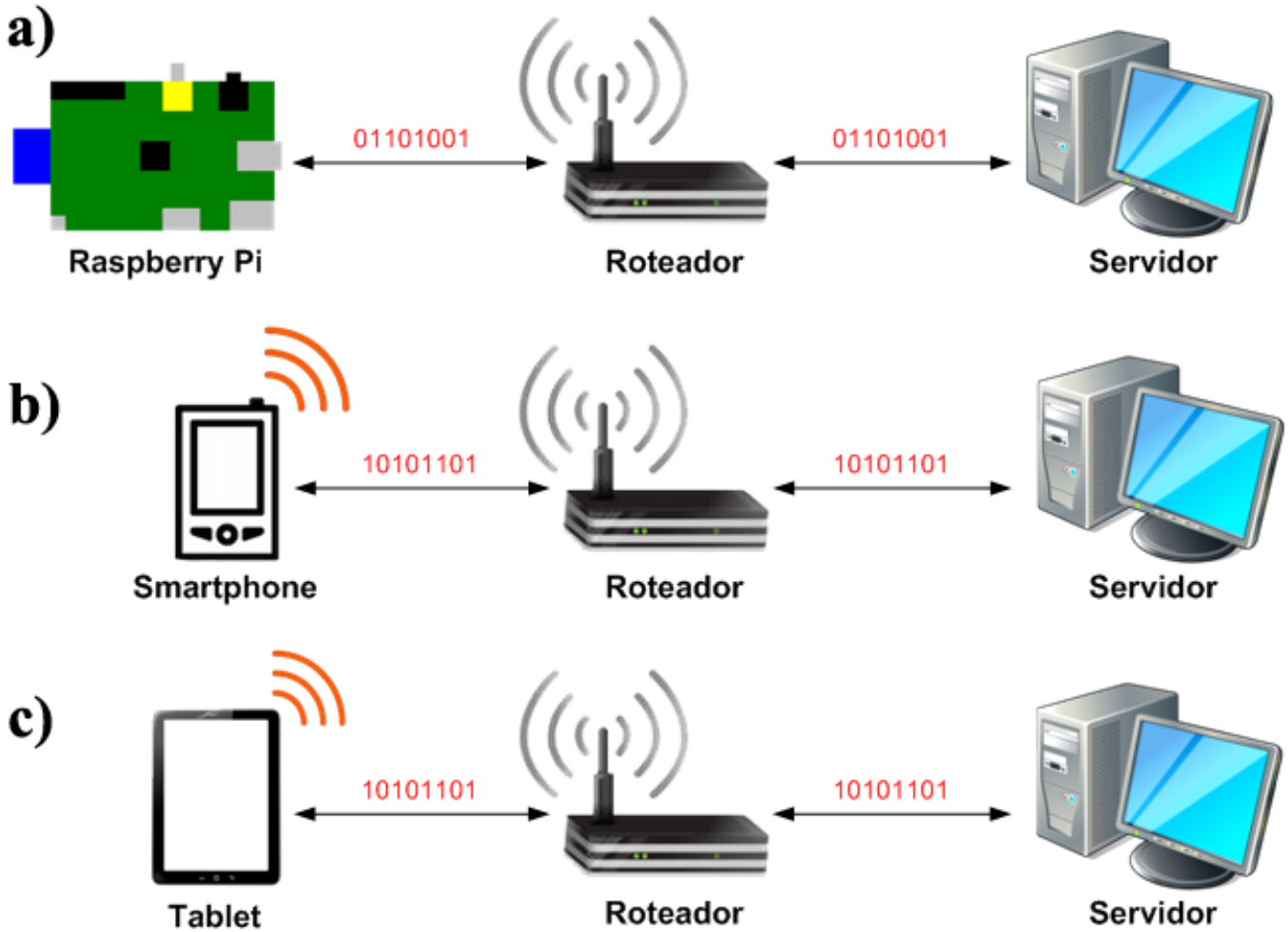



Figura: Comunicação entre o servidor e os dispositivos: **a)** Raspberry Pi (ARMv6), **b)** Motorola Milestone 1 (ARMv7), e **C)** Google Nexus 7 (ARMv7)

# Biblioteca criptográfica *Relic-toolkit*

- Escrita principalmente em linguagem de programação C.
- Suporte para várias arquiteturas de processadores: x86, x86-64, ARM, AVR, MSP.
- Operações sobre números inteiros de precisão arbitrária, corpos finitos, curvas elípticas, emparelhamentos bilineares.
- Outras opções: processamento paralelo (ambientes *multi-core*), diferentes geradores de números pseudoaleatórios, diferentes *timers* (CYCLE, ANSI/POSIX *compatible*, *per-thread*, *per-process*, *realtime*), possibilidade de uso da biblioteca GMP como backend aritmético, diferentes sistemas operacionais, etc.

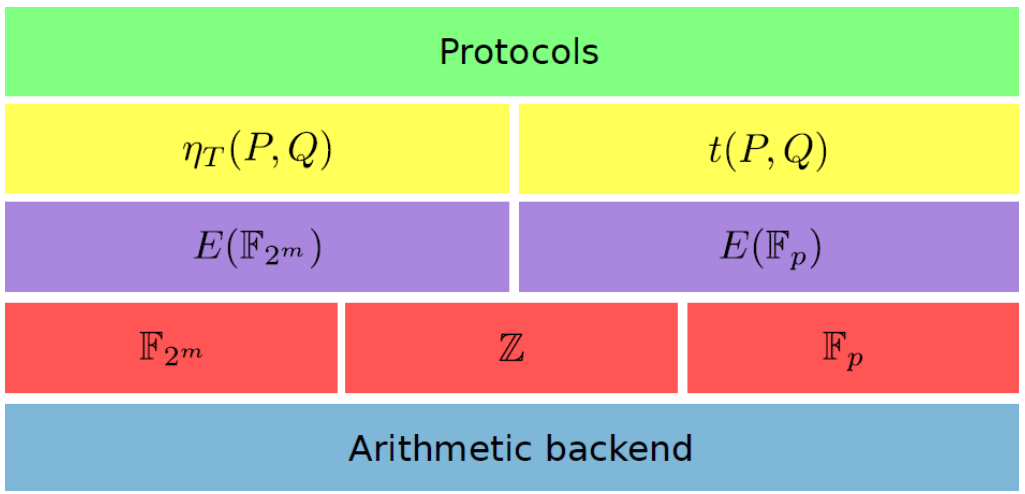



Figura: Esrutura da *Relic-toolkit* [Aranha, 2012]

# Biblioteca criptográfica *Relic-toolkit*

```
/* ------------------------------------------------------------ */
void huang_cao_client(int sock_port){
    char idA[ID_LEN], idB[ID_LEN], server_address[100];
    bn_t rb, order, SK1, SK2;
    g1_t Q1a, Q2a, S1b, S2b;
    g2_t PKm, Ra, Rb, g2_temp1, g2_temp2, g2_temp3, Z3;
    gt_t Z1, Z2;
```

Figura: Declaração de variáveis

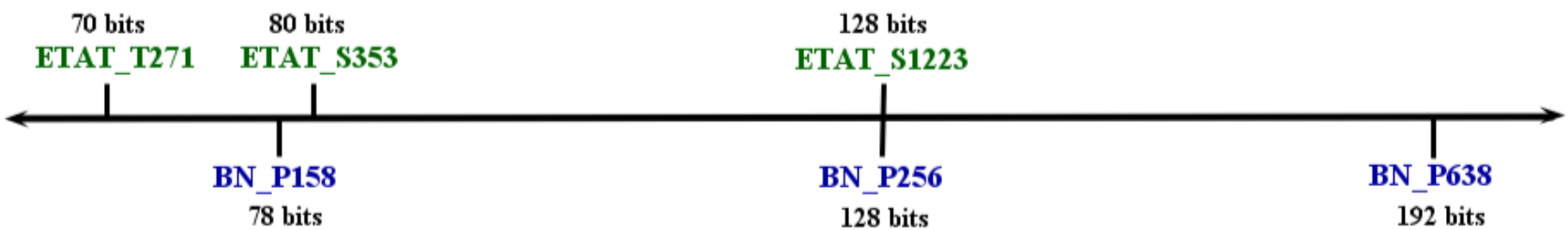
```
g2_mul(g2_temp1, PKm, rb); /*    (rb) * s*P    */
g2_norm(g2_temp1, g2_temp1);

/* Computation of Z1 */
g2_add(g2_temp2, Ra, Q1a); /*   Ra + Q1a    */
g2_add(g2_temp3, g2_temp1, S1b); /*   (rb) * S1b    */
g2_norm(g2_temp2, g2_temp2);
g2_norm(g2_temp3, g2_temp3);
pc_map(Z1, g2_temp2, g2_temp3); /*   e(Ra + Q1a, (rb)*s*P + S1b)   */
```

Figura: Uso de algumas funções

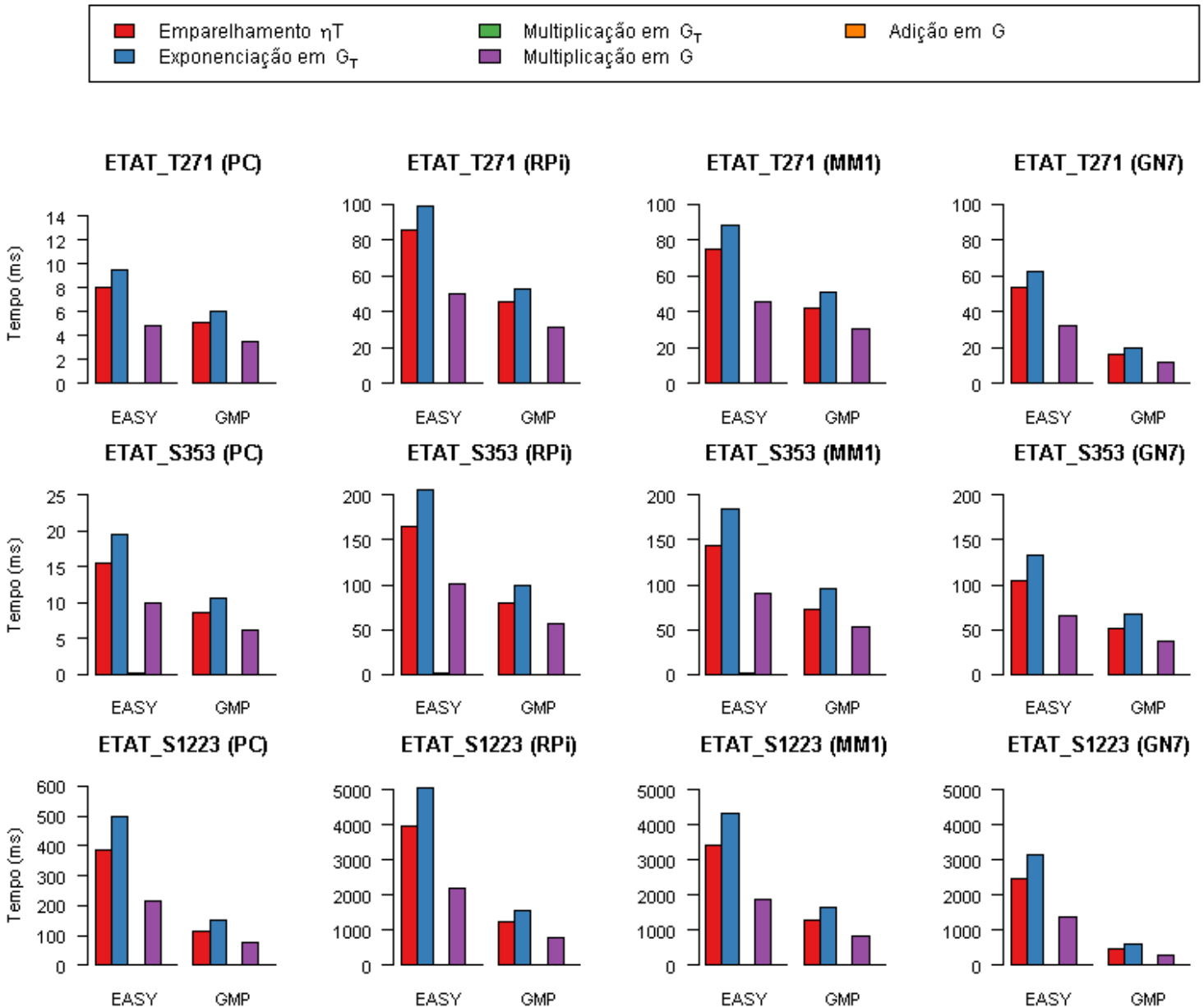
# Curvas Elípticas e Emparelhamentos na *Relic-toolkit*

- **Curvas binárias**
  - ETAT_T271: 70 bits
  - ETAT_S353: 80 bits
  - ETAT_S1223: 128 bits
  - Emparelhamento: $\eta T$

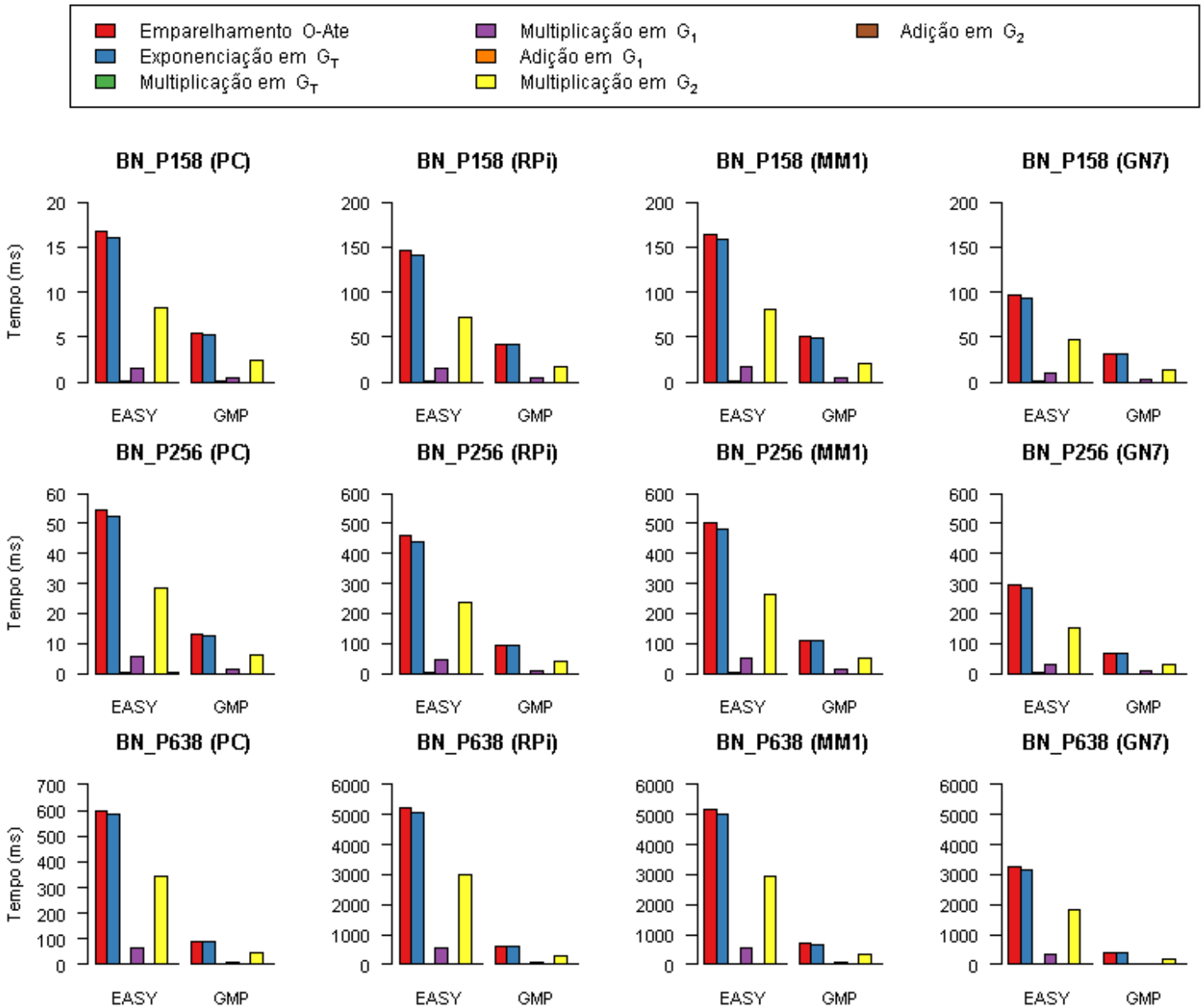
- **Curvas primas**
  - BN_P158: 78 bits
  - BN_P256: 128 bits
  - BN_P638: 192 bits
  - Emparelhamento: O-Ate



# Custo das Operações em Curvas Binárias



# Custo das Operações em Curvas Primas



# Adaptação para Emparelhamento Assimétrico

O PKG publica os seguintes parâmetros:
$\langle k, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, e, P, P', sP, sP', \mathcal{H}_0, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}'_1, \mathcal{H}'_2 \rangle$, onde:

- $k$ é um parâmetro de segurança
- $q$ é a ordem dos grupos $\mathbb{G}_1$ (aditivo) e $\mathbb{G}_T$ (multiplicativo)
- $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$, é um emparelhamento bilinear admissível
- $P$ é um ponto gerador de $\mathbb{G}_1$
- $P'$ é um ponto gerador de $\mathbb{G}_2$
- $\langle sP, sP' \rangle$ é a chave pública do PKG
- $\mathcal{H}_0, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}'_1, \mathcal{H}'_2$ são funções de hash tais que: $\mathcal{H}_0 : \{0,1\}^* \rightarrow \{0,1\}^k$, $\mathcal{H}_1, \mathcal{H}_2 : \{0,1\}^* \rightarrow \mathbb{G}_1$ e $\mathcal{H}'_1, \mathcal{H}'_2 : \{0,1\}^* \rightarrow \mathbb{G}_2$

Cada usuário $U$ possui:

- **Chave pública**: $\left\langle Q_{U_i}, Q'_{U_i} \right\rangle$, tal que $Q_{U_i} = \mathcal{H}_i(ID_U)$ e $Q'_{U_i} = \mathcal{H}'_i(ID_U)$, para $i = \{1, 2\}$
- **Chave privada**: $\left\langle S_{U_i}, S'_{U_i} \right\rangle$, tal que $S_{U_i} = sQ_{U_i}$ e $S'_{U_i} = sQ'_{U_i}$

# Adaptação para Emparelhamento Assimétrico

**Objetivo**: maximizar multiplicações em $\mathbb{G}_1$ e minimizar multiplicações em $\mathbb{G}_2$ para o cliente (participante $B$).

- Multiplicações em $\mathbb{G}_2$ custam aproximadamente 4 vezes mais que em $\mathbb{G}_1$.

# Exemplo: Adaptação para Emparelhamento Assimétrico

Cálculo menos eficiente para o cliente (participante $B$)

| Operação | Quantidade para $A$ | Quantidade para $B$ |
|---|---|---|
| Multiplicação em $\mathbb{G}_1$ | 3 | 0 |
| Multiplicação em $\mathbb{G}_2$ | 1 | 3 |

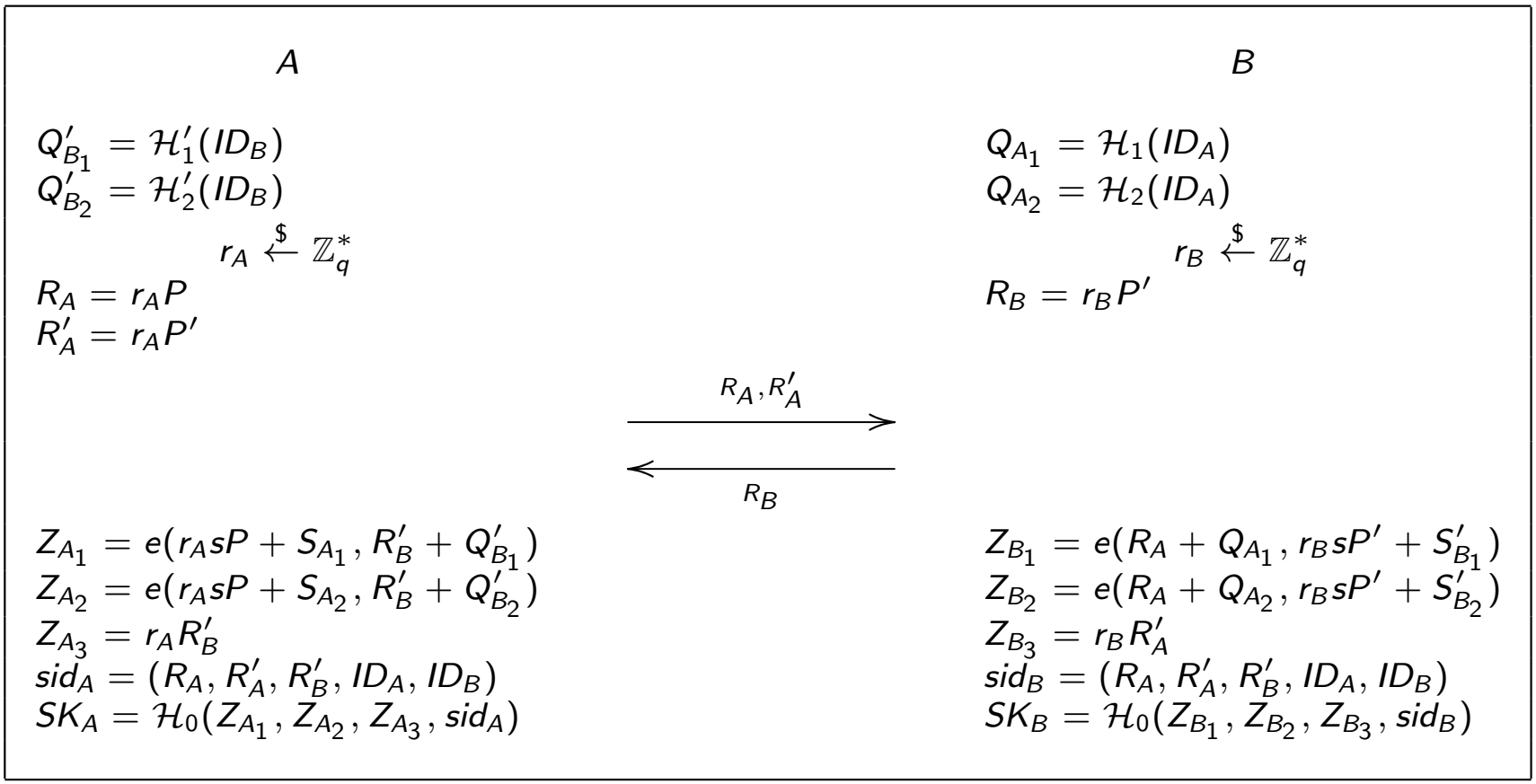



Figura: Protocolo (Huang-Cao, 2009) assimétrico - caso 1

# Exemplo: Adaptação para Emparelhamento Assimétrico

Cálculo um pouco mais eficiente para o cliente (participante $B$)

| Operação | Quantidade para $A$ | Quantidade para $B$ |
|---|---|---|
| Multiplicação em $\mathbb{G}_1$ | 3 | 2 |
| Multiplicação em $\mathbb{G}_2$ | 0 | 2 |

| $A$ | | $B$ |
|---|---|---|
| $Q'_{B_1} = \mathcal{H}'_1(ID_B)$ | | $Q_{A_1} = \mathcal{H}_1(ID_A)$ |
| $Q'_{B_2} = \mathcal{H}'_2(ID_B)$ | | $Q_{A_2} = \mathcal{H}_2(ID_A)$ |
| $r_A \xleftarrow{\$} \mathbb{Z}_q^*$ | | $r_B \xleftarrow{\$} \mathbb{Z}_q^*$ |
| $R_A = r_A P$ | | $R_B = r_B P$ |
| | | $R'_B = r_B P'$ |
| | $R_A \rightarrow$ | |
| | $\leftarrow R_B, R'_B$ | |
| $Z_{A_1} = e(r_A sP + S_{A_1}, R'_B + Q'_{B_1})$ | | $Z_{B_1} = e(R_A + Q_{A_1}, r_B sP' + S'_{B_1})$ |
| $Z_{A_2} = e(r_A sP + S_{A_2}, R'_B + Q'_{B_2})$ | | $Z_{B_2} = e(R_A + Q_{A_2}, r_B sP' + S'_{B_2})$ |
| $Z_{A_3} = r_A R_B$ | | $Z_{B_3} = r_B R_A$ |
| $sid_A = (R_A, R_B, R'_B, ID_A, ID_B)$ | | $sid_B = (R_A, R_B, R'_B, ID_A, ID_B)$ |
| $SK_A = \mathcal{H}_0(Z_{A_1}, Z_{A_2}, Z_{A_3}, sid_A)$ | | $SK_B = \mathcal{H}_0(Z_{B_1}, Z_{B_2}, Z_{B_3}, sid_B)$ |

Figura: Protocolo (Huang-Cao, 2009) assimétrico - caso 2

# Exemplo: Adaptação para Emparelhamento Assimétrico

Cálculo mais eficiente para o cliente (participante $B$)

| Operação | Quantidade para $A$ | Quantidade para $B$ |
|---|---|---|
| Multiplicação em $\mathbb{G}_1$ | 2 | 3 |
| Multiplicação em $\mathbb{G}_2$ | 2 | 0 |

| $A$ | | $B$ |
|---|---|---|
| $Q_{B_1} = \mathcal{H}_1(ID_B)$ | | $Q'_{A_1} = \mathcal{H}'_1(ID_A)$ |
| $Q_{B_2} = \mathcal{H}_2(ID_B)$ | | $Q'_{A_2} = \mathcal{H}'_2(ID_A)$ |
| $r_A \xleftarrow{\$} \mathbb{Z}_q^*$ | | $r_B \xleftarrow{\$} \mathbb{Z}_q^*$ |
| $R_A = r_A P$ | | $R_B = r_B P$ |
| $R'_A = r_A P'$ | | |
| | $\xrightarrow{R_A, R'_A}$ | |
| | $\xleftarrow{R_B}$ | |
| $Z_{A_1} = e(R_B + Q_{B_1}, r_A s P' + S'_{A_1})$ | | $Z_{B_1} = e(r_B s P + S_{B_1}, R'_A + Q'_{A_1})$ |
| $Z_{A_2} = e(R_B + Q_{B_2}, r_A s P' + S'_{A_2})$ | | $Z_{B_2} = e(r_B s P + S_{B_2}, R'_A + Q'_{A_2})$ |
| $Z_{A_3} = r_A R_B$ | | $Z_{B_3} = r_B R_A$ |
| $sid_A = (R_A, R'_A, R_B, ID_A, ID_B)$ | | $sid_B = (R_A, R'_A, R_B, ID_A, ID_B)$ |
| $SK_A = \mathcal{H}_0(Z_{A_1}, Z_{A_2}, Z_{A_3}, sid_A)$ | | $SK_B = \mathcal{H}_0(Z_{B_1}, Z_{B_2}, Z_{B_3}, sid_B)$ |

Figura: Protocolo (Huang-Cao, 2009) assimétrico - caso 3

# Roteiro

1 Introdução

2 Modelos Alternativos

3 Protocolos

4 Ambiente de Testes

5 Resultados

6 Conclusão

# Quantidade de Operações

| Protocolos *ID-Based* | Pré-computação | $e(,)$ | $E_{\mathbb{G}_T}$ | $M_{\mathbb{G}_T}$ | $M_{\mathbb{G}}$ | $A_{\mathbb{G}}$ |
|---|---|---|---|---|---|---|
| HC-BDH | Não | 2 | 0 | 0 | 2 | 4 |
| | Sim | 2 | 0 | 0 | 2 | 4 |
| HLZ-GBDH | Não | 2 | 0 | 0 | 1 | 2 |
| | Sim | 2 | 0 | 0 | 1 | 2 |
| CC-BDH | Não | 1 | 0 | 0 | 3 | 2 |
| | Sim | 1 | 0 | 0 | 3 | 2 |
| NCLH-BDH | Não | 6 | 2 | 2 | 1 | 0 |
| | Sim | 2 | 2 | 2 | 1 | 0 |
| NCLH-GBDH | Não | 3 | 1 | 1 | 1 | 0 |
| | Sim | 1 | 1 | 1 | 1 | 0 |
| NCL-BDH | Não | 2 | 0 | 0 | 1 | 4 |
| | Sim | 2 | 0 | 0 | 1 | 4 |
| Protocolos *Certificateless* | Pré-computação | $e(,)$ | $E_{\mathbb{G}_T}$ | $M_{\mathbb{G}_T}$ | $M_{\mathbb{G}}$ | $A_{\mathbb{G}}$ |
| LBG-BDH | Não | 10 | 0 | 4 | 6 | 0 |
| | Sim | 4 | 0 | 2 | 4 | 0 |
| LBG-GBDH | Não | 5 | 0 | 2 | 6 | 0 |
| | Sim | 2 | 0 | 1 | 4 | 0 |
| GOT-BDH | Não | 8 | 0 | 2 | 6 | 4 |
| | Sim | 2 | 0 | 0 | 4 | 4 |
| GOT-GBDH | Não | 4 | 0 | 1 | 6 | 2 |
| | Sim | 1 | 0 | 0 | 4 | 2 |
| GNT1-GBDH | Não | 4 | 0 | 1 | 6 | 4 |
| | Sim | 2 | 0 | 0 | 4 | 4 |
| GNT3-BDH | Não | 6 | 0 | 2 | 6 | 4 |
| | Sim | 2 | 0 | 0 | 4 | 4 |
| GNT2-GBDH | Não | 3 | 0 | 0 | 6 | 4 |
| | Sim | 1 | 0 | 0 | 4 | 2 |
| GNT4-GBDH | Não | 4 | 0 | 0 | 6 | 8 |
| | Sim | 2 | 0 | 0 | 4 | 4 |

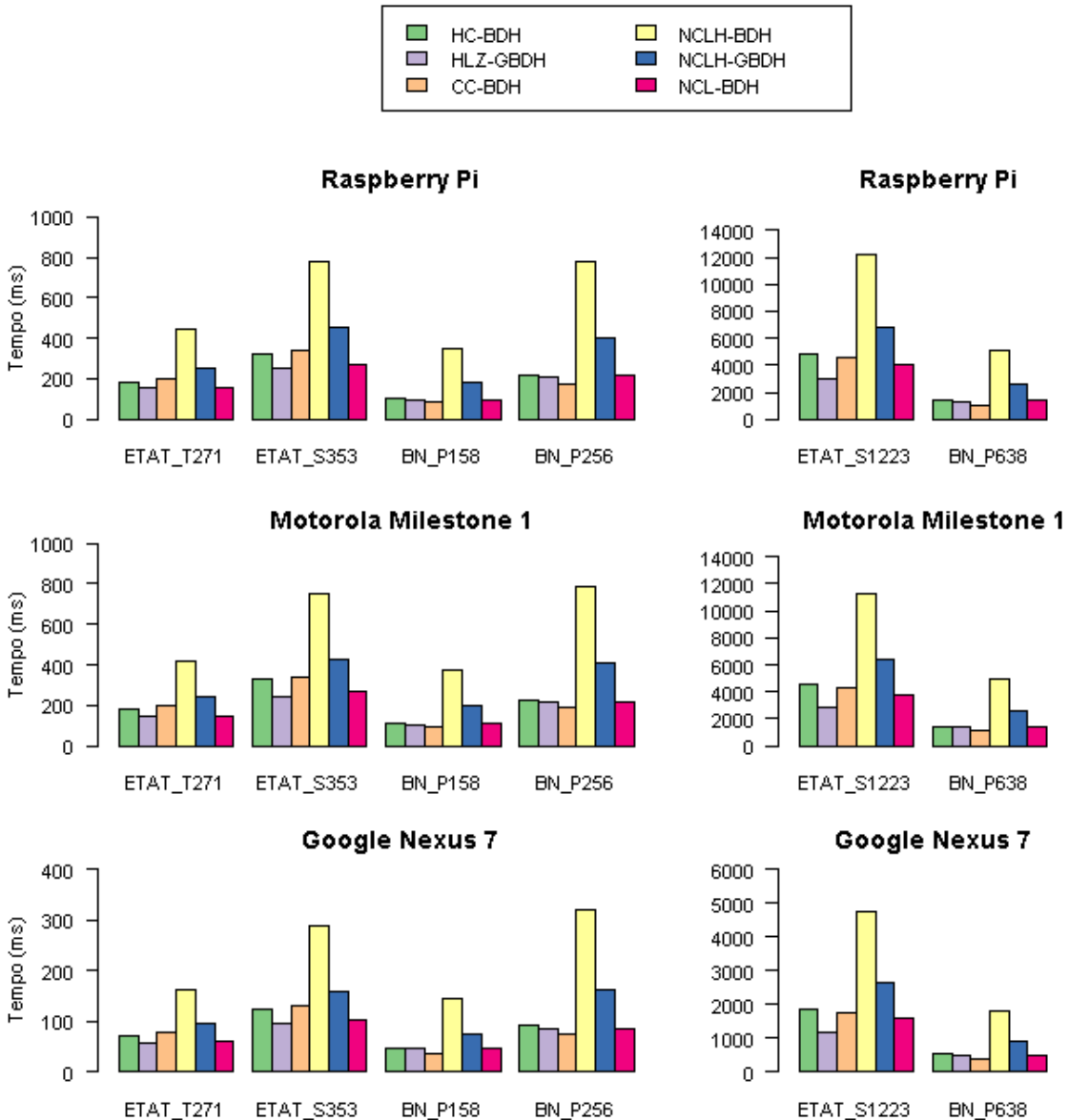
Tabela: Quantidade de operações por protocolos (emparelhamento simétrico)

# Quantidade de Operações

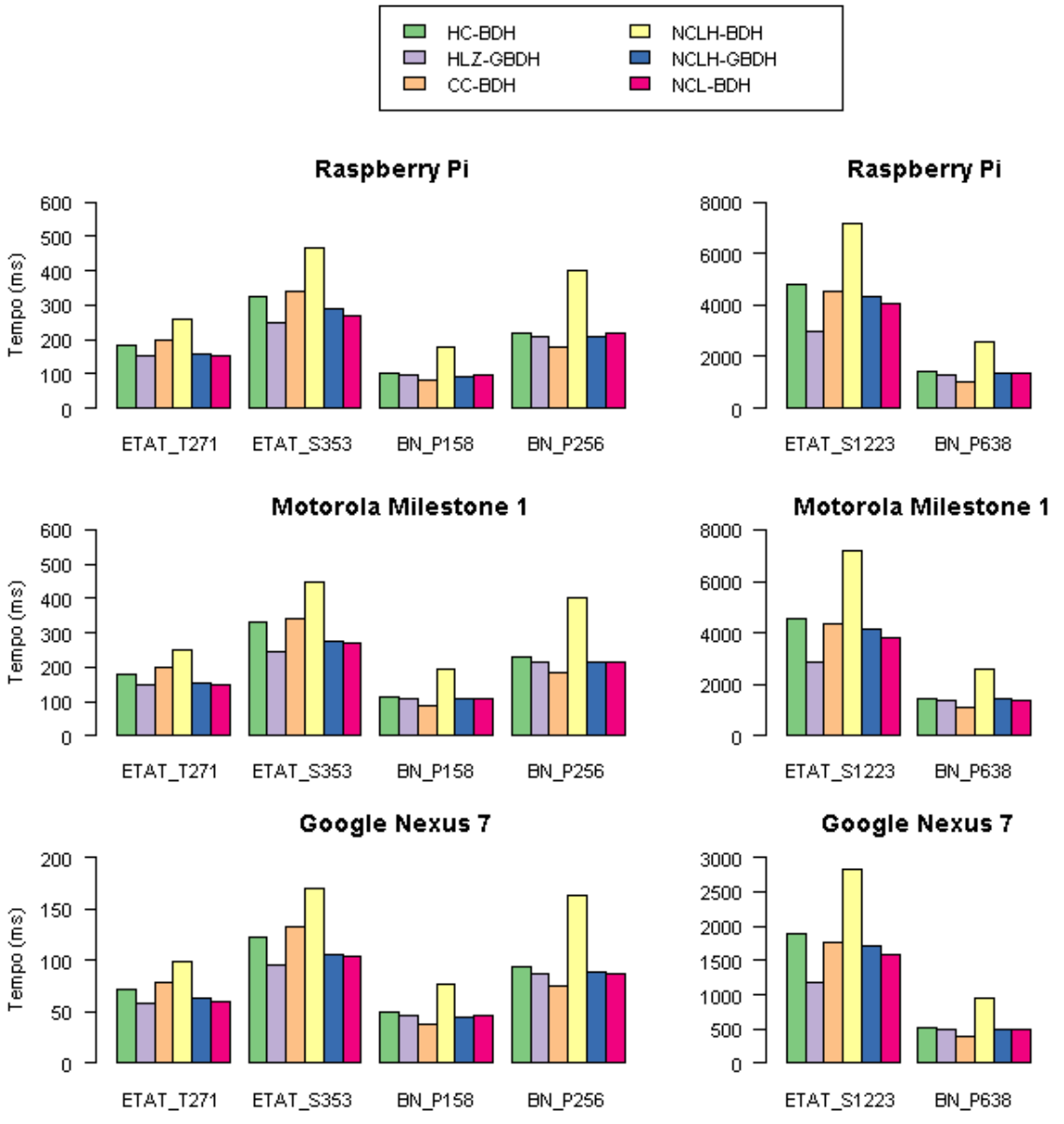
| Protocolos *ID-Based* | Pré-computação | $e(,)$ | $E_{\mathbb{G}_T}$ | $M_{\mathbb{G}_T}$ | $M_{\mathbb{G}_1}$ | $A_{\mathbb{G}_1}$ | $M_{\mathbb{G}_2}$ | $A_{\mathbb{G}_2}$ |
|---|---|---|---|---|---|---|---|---|
| HC-BDH | Não | 2 | 0 | 0 | 2 | 2 | 0 | 2 |
| | Sim | 2 | 0 | 0 | 2 | 2 | 0 | 2 |
| HLZ-GBDH | Não | 2 | 0 | 0 | 1 | 1 | 0 | 1 |
| | Sim | 2 | 0 | 0 | 1 | 1 | 0 | 1 |
| CC-BDH | Não | 1 | 0 | 0 | 2 | 0 | 1 | 1 |
| | Sim | 1 | 0 | 0 | 2 | 0 | 1 | 1 |
| NCLH-BDH | Não | 6 | 2 | 2 | 1 | 0 | 0 | 0 |
| | Sim | 2 | 2 | 2 | 1 | 0 | 0 | 0 |
| NCLH-GBDH | Não | 3 | 1 | 1 | 1 | 0 | 0 | 0 |
| | Sim | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| NCL-BDH | Não | 2 | 0 | 0 | 1 | 2 | 0 | 2 |
| | Sim | 2 | 0 | 0 | 1 | 2 | 0 | 2 |
| Protocolos *Certificateless* | Pré-computação | $e(,)$ | $E_{\mathbb{G}_T}$ | $M_{\mathbb{G}_T}$ | $M_{\mathbb{G}_1}$ | $A_{\mathbb{G}_1}$ | $M_{\mathbb{G}_2}$ | $A_{\mathbb{G}_2}$ |
| LBG-BDH | Não | 10 | 0 | 4 | 6 | 0 | 0 | 0 |
| | Sim | 4 | 0 | 2 | 4 | 0 | 0 | 0 |
| LBG-GBDH | Não | 5 | 0 | 2 | 6 | 0 | 0 | 0 |
| | Sim | 2 | 0 | 1 | 4 | 0 | 0 | 0 |
| GOT-BDH | Não | 8 | 0 | 2 | 6 | 2 | 0 | 2 |
| | Sim | 2 | 0 | 0 | 4 | 2 | 0 | 2 |
| GOT-GBDH | Não | 4 | 0 | 1 | 6 | 1 | 0 | 1 |
| | Sim | 1 | 0 | 0 | 4 | 1 | 0 | 1 |
| GNT1-GBDH | Não | 4 | 0 | 1 | 6 | 2 | 0 | 2 |
| | Sim | 2 | 0 | 0 | 4 | 2 | 0 | 2 |
| GNT3-BDH | Não | 6 | 0 | 0 | 4 | 2 | 0 | 2 |
| | Sim | 2 | 0 | 0 | 4 | 2 | 0 | 2 |
| GNT2-GBDH | Não | 3 | 0 | 0 | 6 | 2 | 0 | 2 |
| | Sim | 1 | 0 | 0 | 4 | 1 | 0 | 1 |
| GNT4-GBDH | Não | 4 | 0 | 0 | 6 | 4 | 0 | 4 |
| | Sim | 2 | 0 | 0 | 4 | 2 | 0 | 2 |

Tabela: Quantidade de operações por protocolos (emparelhamento assimétrico)

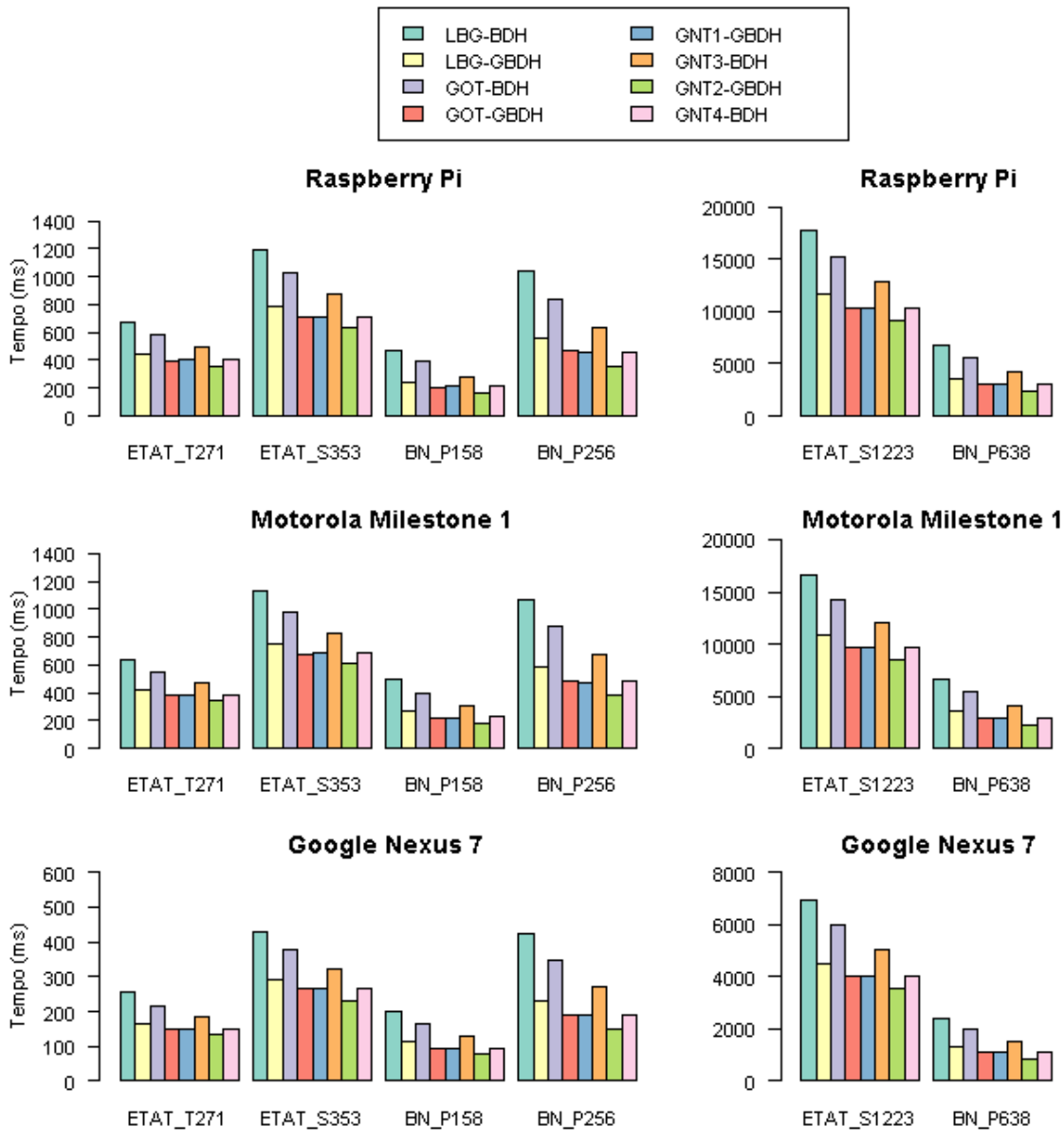
# Medidas de Tempo: protocolos *ID-based* sem pré-computação



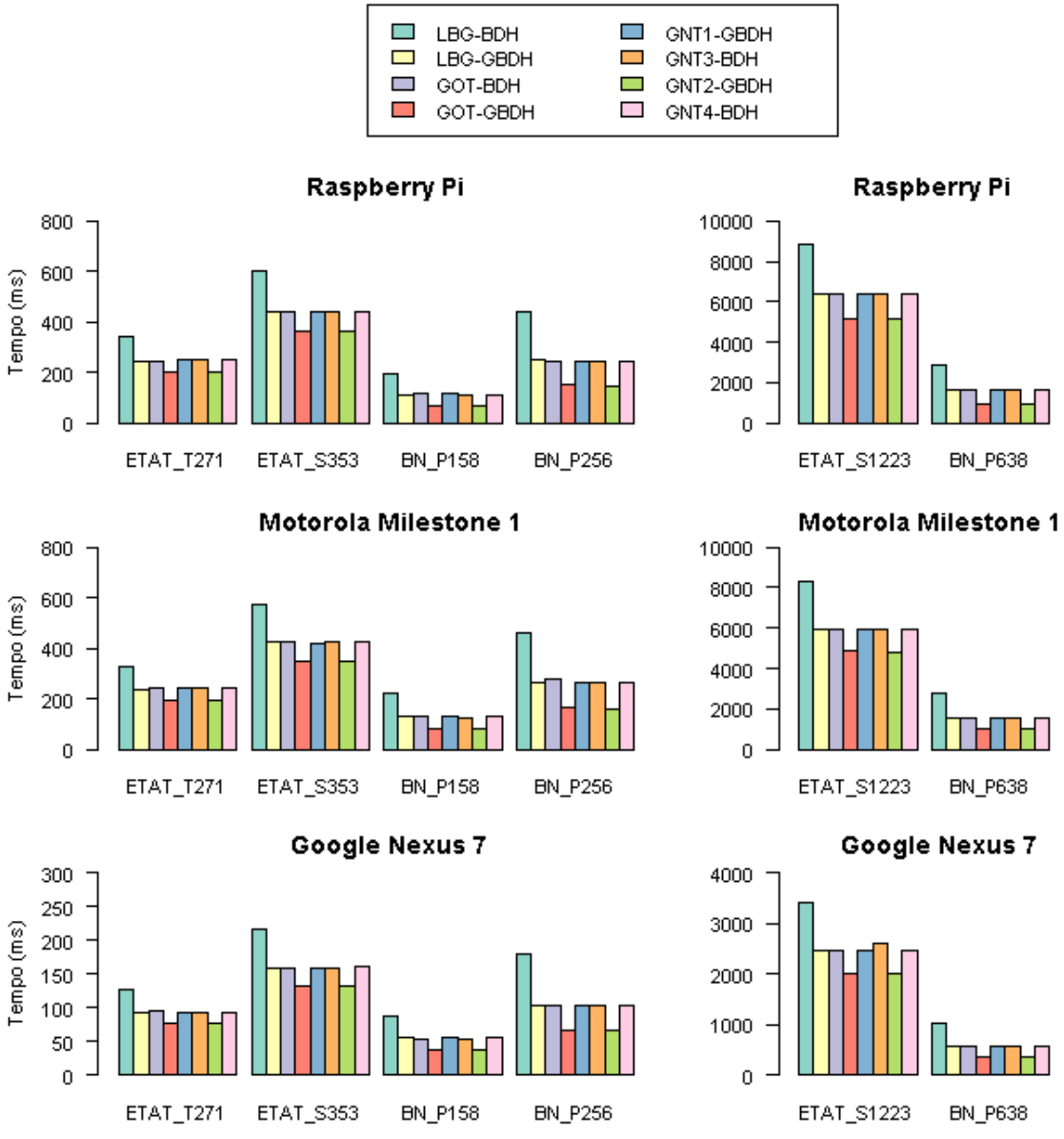
# Medidas de Tempo: protocolos *ID-based* com pré-computação



# Medidas de Tempo: protocolos *Certificateless* sem pré-computação



# Medidas de Tempo: protocolos *Certificateless* com pré-computação



# Uso de Memória

| | Curva Elíptica | | | | | |
|---|---|---|---|---|---|---|
| Protocolos *ID-Based* | ETAT_T271 | ETAT_S353 | ETAT_S1223 | BN_P158 | BN_P256 | BN_P638 |
| HC-BDH | 1704 | 2244 | 7120 | 1624 | 2536 | 6184 |
| HLZ-GBDH | 1460 | 1916 | 6044 | 1376 | 2132 | 5156 |
| CC-BDH | 1612 | 2104 | 6572 | 1460 | 2228 | 5300 |
| NCLH-BDH | 2232 | 2952 | 9448 | 2812 | 4444 | 10972 |
| NCLH-GBDH | 1576 | 2080 | 6632 | 1904 | 2996 | 7364 |
| NCL-BDH | 1592 | 2096 | 6648 | 1496 | 2336 | 5696 |
| Protocolos *Certificateless* | ETAT_T271 | ETAT_S353 | ETAT_S1223 | BN_P158 | BN_P256 | BN_P638 |
| LBG-BDH | 2884 | 3808 | 12148 | 3324 | 5232 | 12864 |
| LBG-GBDH | 2228 | 2936 | 9332 | 2416 | 3784 | 9256 |
| GOT-BDH | 2884 | 3808 | 12148 | 3448 | 5428 | 13348 |
| GOT-GBDH | 2228 | 2936 | 9332 | 2540 | 3980 | 9740 |
| GNT1-GBDH | 2452 | 3232 | 10276 | 2728 | 4276 | 10468 |
| GNT3-BDH | 2596 | 3424 | 10900 | 2968 | 4660 | 11428 |
| GNT2-GBDH | 1940 | 2552 | 8084 | 2060 | 3212 | 7820 |
| GNT4-BDH | 2308 | 3040 | 9652 | 2488 | 3892 | 9508 |

Tabela: Uso de memória (em *bytes*) dos protocolos analisados para diferentes níveis de segurança

# Comprimento das Mensagens Trocadas

| | Curva Elíptica | | | | | |
|---|---|---|---|---|---|---|
| Protocolos *ID-Based* | ETAT_T271 | ETAT_S353 | ETAT_S1223 | BN_P158 | BN_P256 | BN_P638 |
| HC-BDH | 224 | 296 | 944 | 252 | 396 | 972 |
| HLZ-GBDH | 224 | 296 | 944 | 188 | 296 | 728 |
| CC-BDH | 448 | 592 | 1888 | 316 | 496 | 1216 |
| NCLH-BDH | 224 | 296 | 944 | 252 | 396 | 972 |
| NCLH-GBDH | 224 | 296 | 944 | 252 | 396 | 972 |
| NCL-BDH | 224 | 296 | 944 | 188 | 296 | 728 |
| Protocolos *Certificateless* | ETAT_T271 | ETAT_S353 | ETAT_S1223 | BN_P158 | BN_P256 | BN_P638 |
| LBG-BDH | 448 | 592 | 1888 | 504 | 792 | 1944 |
| LBG-GBDH | 448 | 592 | 1888 | 504 | 792 | 1944 |
| GOT-BDH | 448 | 592 | 1888 | 504 | 792 | 1944 |
| GOT-GBDH | 448 | 592 | 1888 | 504 | 792 | 1944 |
| GNT1-GBDH | 448 | 592 | 1888 | 504 | 792 | 1944 |
| GNT3-BDH | 448 | 592 | 1888 | 504 | 792 | 1944 |
| GNT2-GBDH | 448 | 592 | 1888 | 504 | 792 | 1944 |
| GNT4-BDH | 448 | 592 | 1888 | 504 | 792 | 1944 |

Tabela: Comprimento das mensagens trocadas (em *bytes*) dos protocolos analisados para diferentes níveis de segurança

# Roteiro

1 Introdução

2 Modelos Alternativos

3 Protocolos

4 Ambiente de Testes

5 Resultados

6 Conclusão

# Sugestões para Trabalhos Futuros

- Paralelização em ambientes *multi-core*.
  - Há suporte para OpenMP na plataforma Android?
- Testes com a biblioteca GMP, versões 5.1.x em diante.
  - Otimizações para processadores ARM, principalmente Cortex-A9.
- Experimentos com novas versões da *Relic-toolkit* (mais otimizadas) e outras bibliotecas criptográficas.
  - A versão 0.3.1 foi a única que funcionou sem erros para todos os casos.

# Referências Bibliográficas


- AL-RIYAMI. S, PATERSON. K. **Certificateless Public Key Cryptography**. ASIACRYPT 2003.
- ARANHA, D. **Efficient Binary Field Arithmetic and Applications to Curve-based Cryptography**. Tutorial. CHES 2012.
- BONEH. D, FRANKLIN. M. **Identity based encryption from the Weil pairing**. Proceedings of CRYPTO 2001.
- CESENA, E.**Trace Zero Varieties in Pairing-based Cryptography**. Tese de doutorado. Università degli studi Roma Tre. 2010.
- CHOO, K. CHOW, S. **Strongly-secure identity-based key agreement and anonymous extension**. Springer 2007.
- DIFFIE, W. HELLMAN, M. **New Directions in Cryptography**. Information Theory, IEEE Transactions on, v. 22, n. 6, p. 644-654, 1976.
- GOYA, D. **Criptografia de chave pública sem certificado**. Tese de doutorado. Universidade de São Paulo. 2011.

# Referências Bibliográficas


- GOYA, D. NAKAMURA, D. TERADA, R. **Acordo de chave seguro contra autoridade mal intencionada**. SBSeg 2011.
- GOYA, D. OKIDA, C. TERADA, R. **Acordo de chave com autenticação sem certificado digital**. I2TS, 2010.
- HU, X. LIU, W. ZHANG, J. **An Efficient ID-Based Authenticated Key Exchange Protocol**. Information Engineering, 2009. ICIE'09. WASE International Conference on. Vol. 2. IEEE, 2009.
- HUANG, H. CAO, Z. **An ID-based Authenticated Key Exchange Protocol Based on Bilinear Diffie-Hellman Problem**. ASIACCS 2009.
- LIPPOLD, G. BOYD, C. GONZALEZ NIETO, J. **Strongly secure certificateless key agreement**. Pairing 2009.
- NI, L. CHEN, G. LI, J. **Escrowable identity-based authenticated key agreement protocol with strong security**. Computers & Mathematics with Applications. 2012.
- NI, L. CHEN, G. LI, J. HAO, Y. **Strongly secure identity-based authenticated key agreement protocols**. Computers & Electrical Engineering. 2011.

# Perguntas?

Obrigado!

http://www.ime.usp.br/~rwill