IME - Instituto de Matemática e Estatística

Implementando protocolos de acordo de chave baseados em identidade e sem certificado na plataforma Android

Rafael Will Macedo de Araujo
rwill@ime.usp.br
http://www.ime.usp.br/∼rwill

DCC – IME – USP

11 de Setembro de 2012

CNPq no. 151134/2010-3

IME - Instituto de Matemática e Estatística

Objetivos

- Revisão dos conceitos de criptografia de chave pública baseada em identidades e sem certificado.
- Introduzir as ferramentas básicas para construção de código nativo na plataforma *Android*.
- Expor nossa proposta de pesquisa.
- Mostrar os resultados parciais obtidos.

Protocolos de Acordo de Chave

Problema

Como duas pessoas podem combinar uma chave secreta sem necessidade de um encontro?

Solução

Através de um protocolo de acordo de chaves.

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

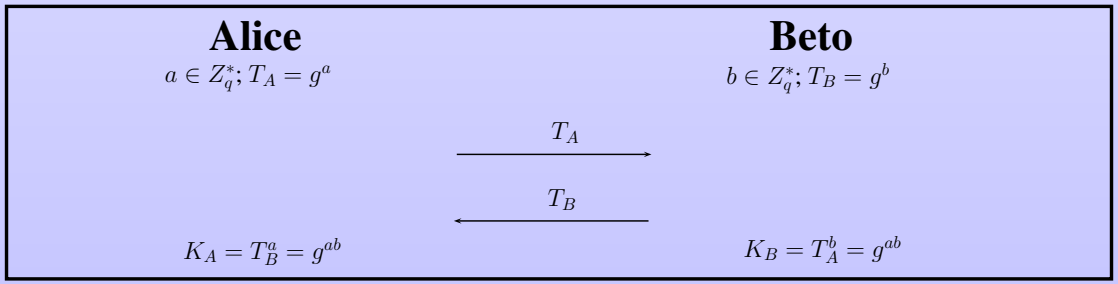


Figura: Acordo de chave Diffie-Hellman (1976)

IME - Instituto de Matemática e Estatística

Criptografia de Chave Pública Baseada em Identidade

- Modelo proposto em 1984, por Adi Shamir.
- A chave pública deixa de ser um valor gerado aleatoriamente, e passa ser a própria **identidade do usuário**.
  - Exemplos: *nome, e-mail, CPF, n° de telefone, endereço IP, etc.*
- Pressupõe a existência de um **Gerador de Chaves Privadas** (PKG - *Private Key Generator*).
  - Responsável por: gerar e guardar a chave-mestra secreta, calcular chaves privadas de todos os usuários, entregar as chaves privadas dos usuários de forma segura.

Diagrama do Modelo Baseado em Identidade

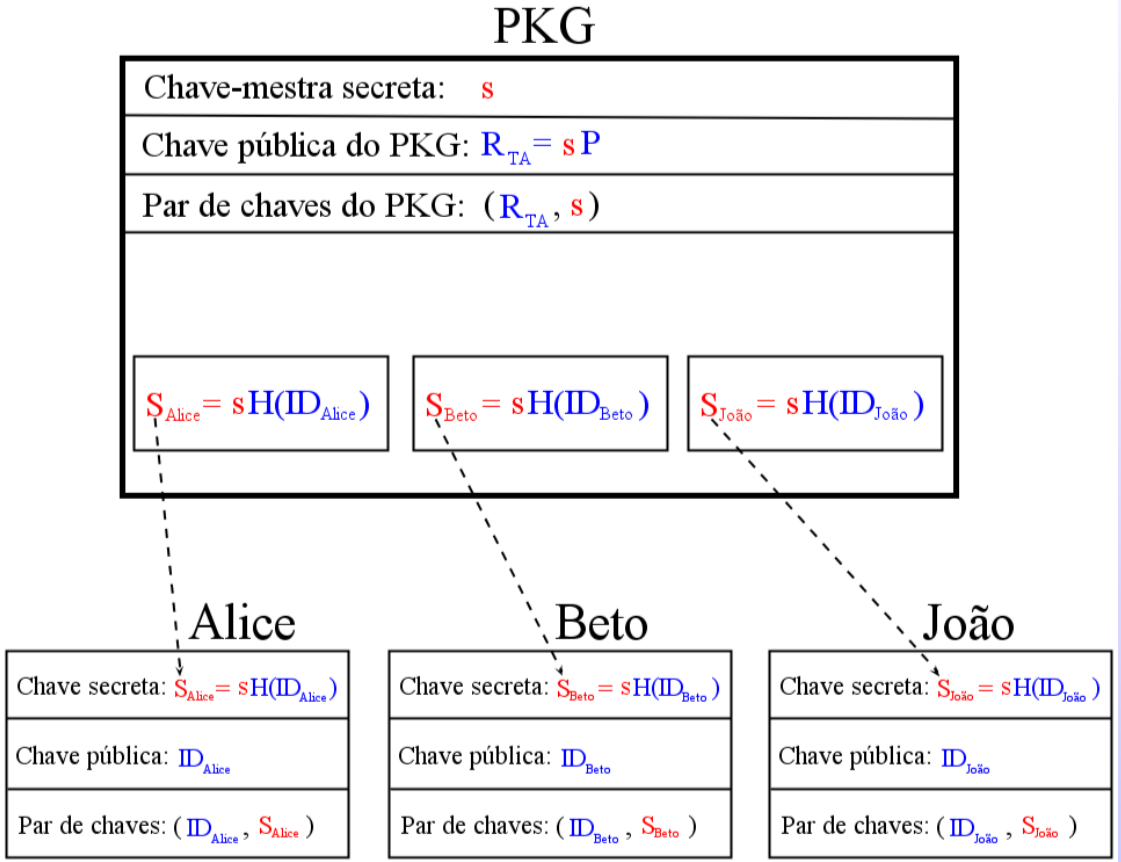


Figura: Distribuição das chaves privadas no modelo baseado em identidade

Criptografia de Chave Pública sem Certificado

- Modelo proposto em 2003 por S. Al-Riyami e K. Paterson.
- Combina as ideias do modelo baseado em identidades com o modelo de chave auto-certificada (proposto por Girault em 1991).
  - Um dos principais objetivos era eliminar a propriedade de custódia de chaves, inerente ao modelo baseado em identidade.
- Trata-se de um modelo intermediário entre o baseado em identidade e a ICP.
- Pressupõe a existência de uma Autoridade de Confiança, chamada de KGC (*Key Generation Center*).
  - Responsável por: gerar e guardar a chave-mestra secreta, calcular chaves secretas parciais de todos os usuários, entregar as chaves secretas parciais dos usuários de forma segura.

Diagrama do Modelo sem Certificado

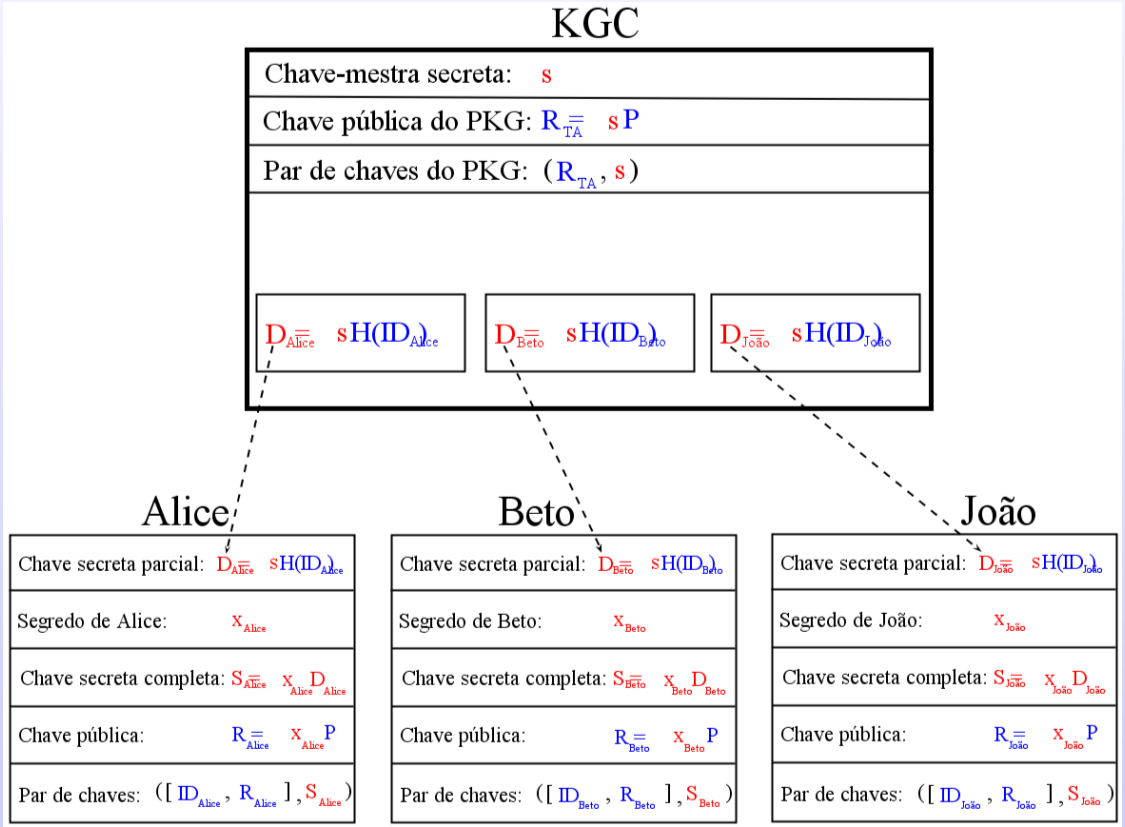


Figura: Distribuição das chaves privadas parciais no modelo sem certificado

IME - Instituto de Matemática e Estatística

Android

- Sistema operacional baseado no núcleo *Linux*, para dispositivos móveis.
- Desenvolvido pelo *Google* e pela *Open Handset Alliance*.
- São disponibilizados dois kits de desenvolvimento para *Android*: **SDK** e **NDK**.

Kits de Desenvolvimento

- *Software Development Kit* (SDK): fornece bibliotecas de API e ferramentas de desenvolvimento necessárias para construir, testar e debugar aplicativos para *Android*. Possibilita a construção de aplicativos em linguagem Java.
- *Native Development Kit* (NDK): permite construir aplicativos ou parte de aplicativos utilizando linguagens de código nativo, como C e C++.
  - Nem sempre é vantajoso escrever parte de um aplicativo em código nativo.

Android SDK

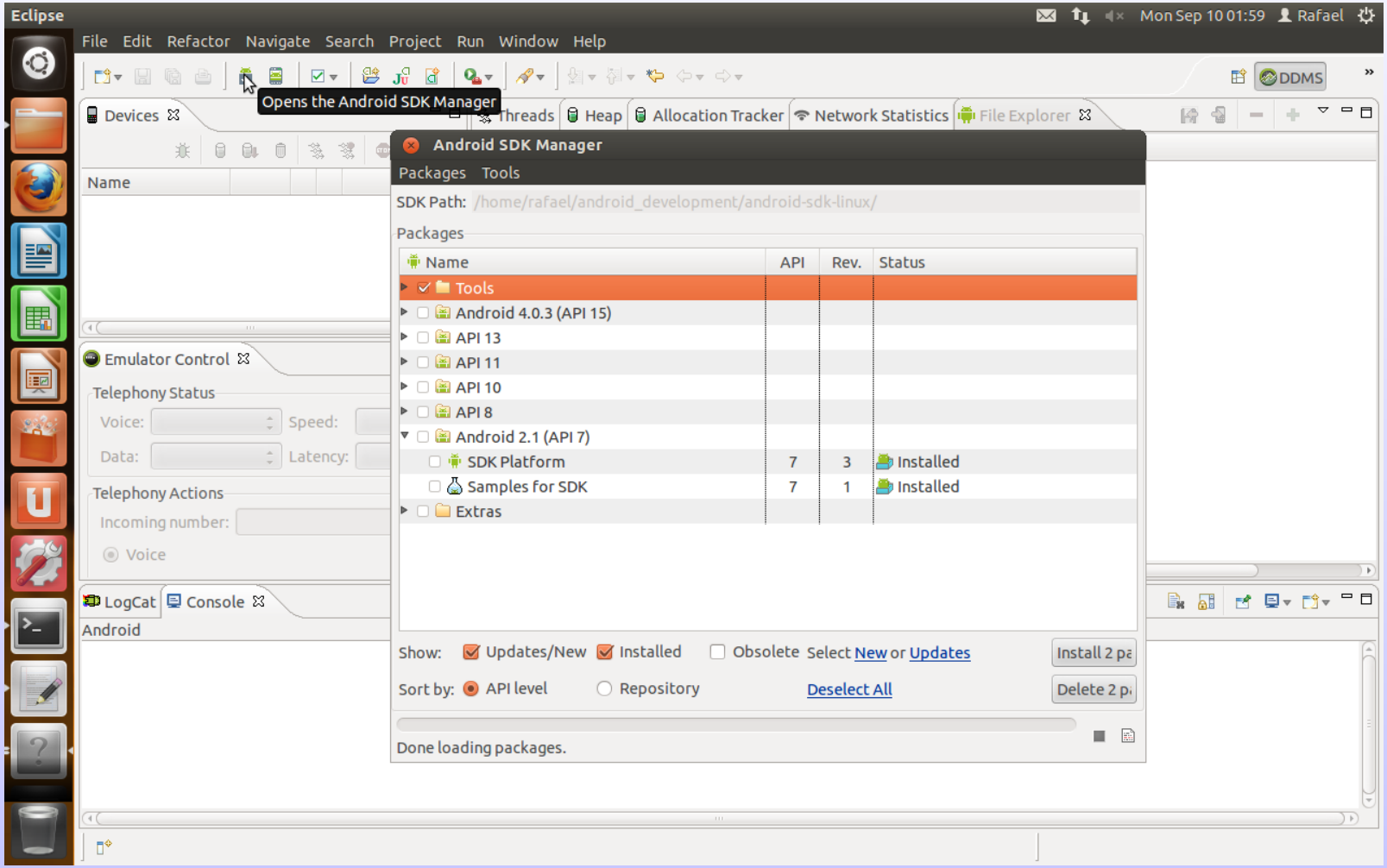


Figura: Android *Software Development Kit Manager*

Android AVD

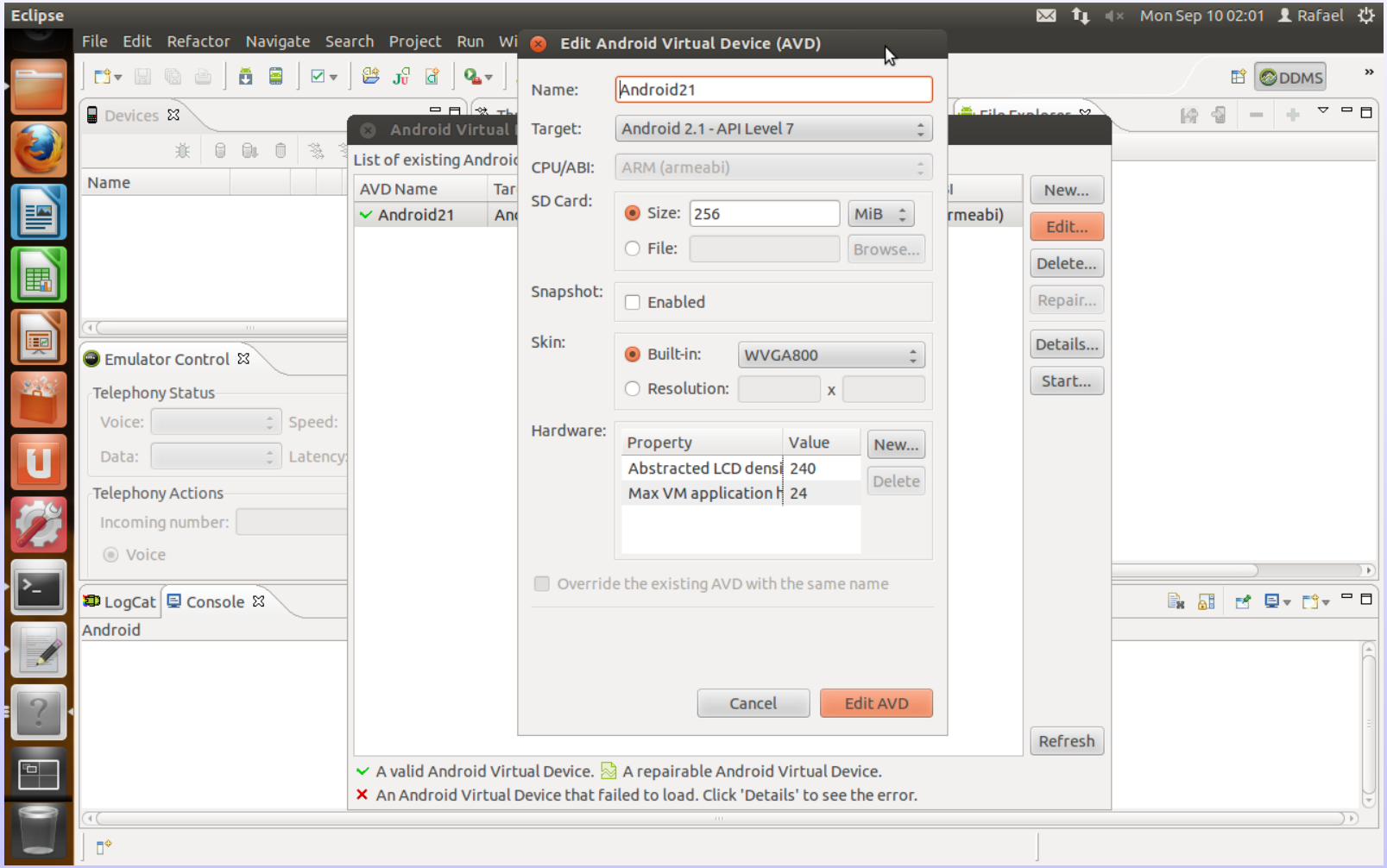


Figura: Android *Virtual Device Manager*

Android *Emulator*

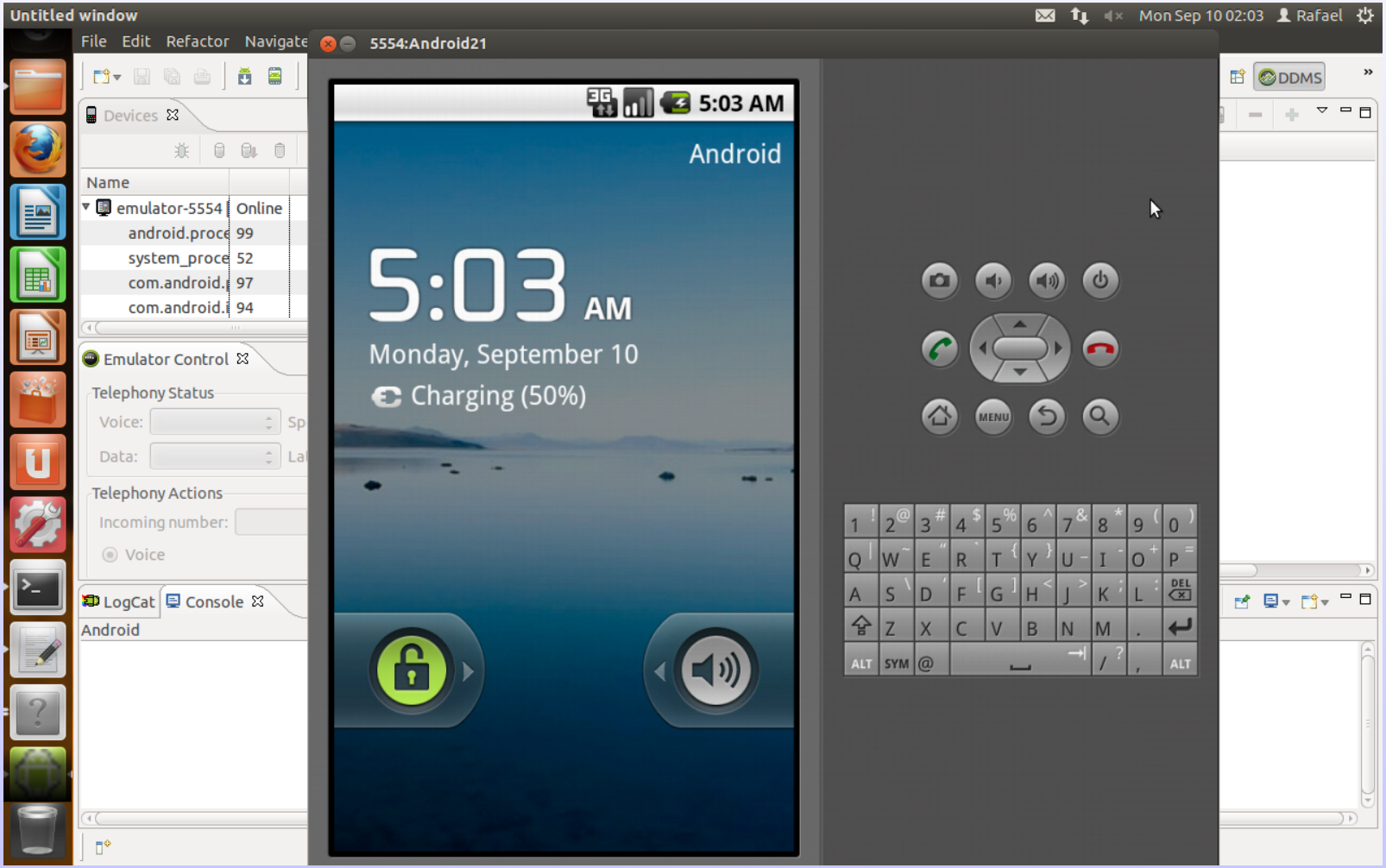


Figura: Emulador de *smartphone*

Android *Emulator*

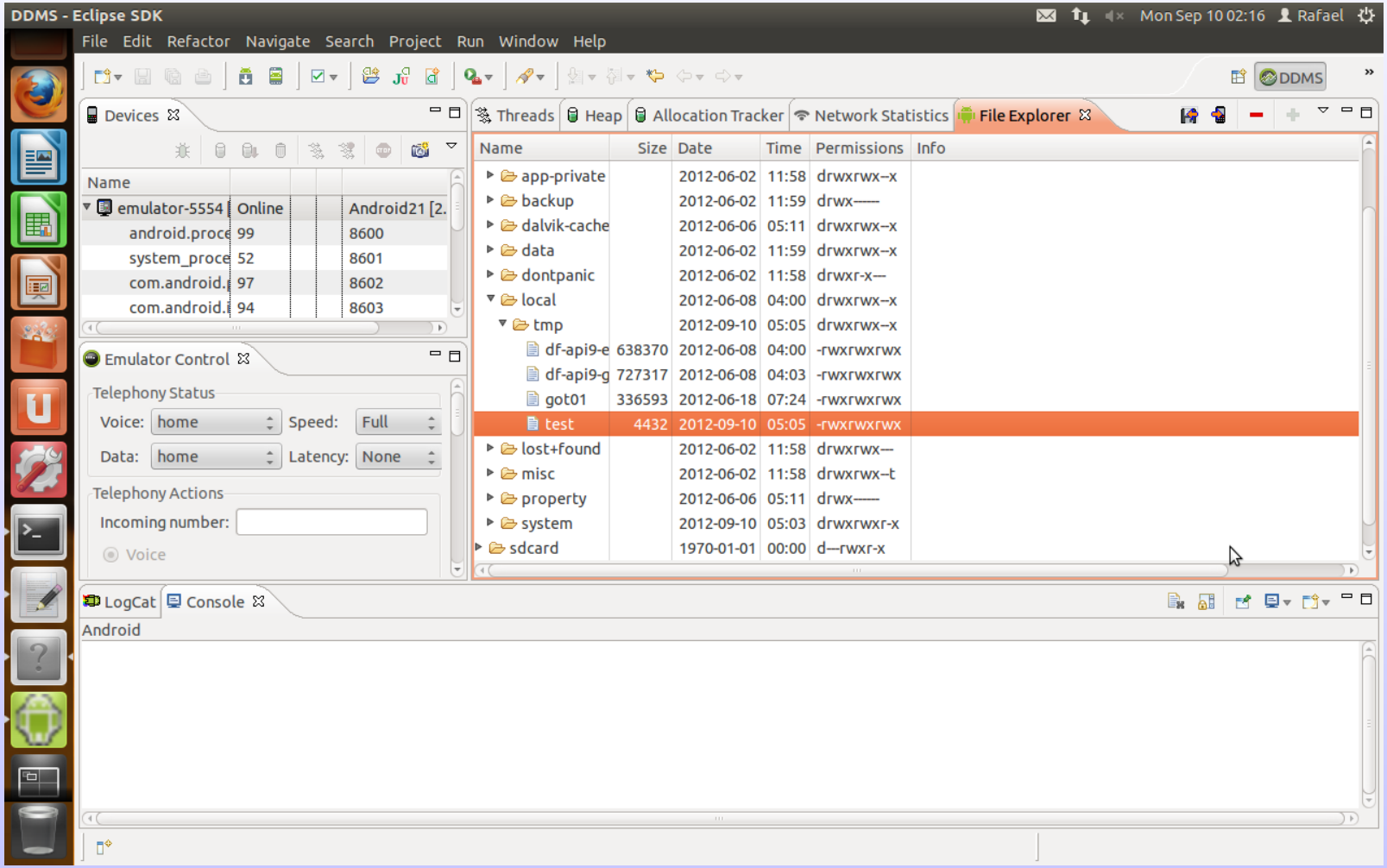


Figura: *File Explorer*

Android *Terminal Emulator*

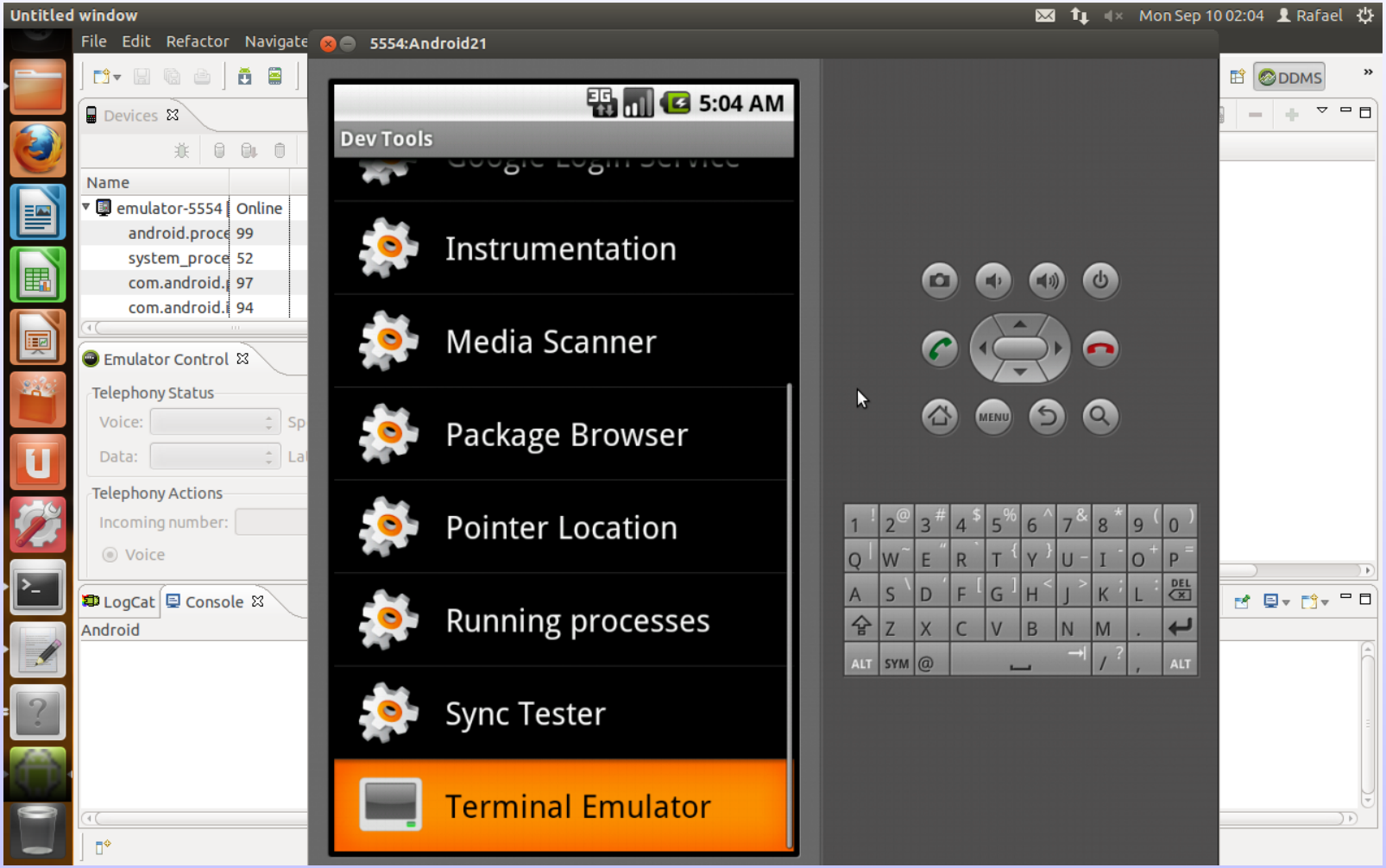


Figura: *Android Virtual Device Manager*

Android *Terminal Emulator*

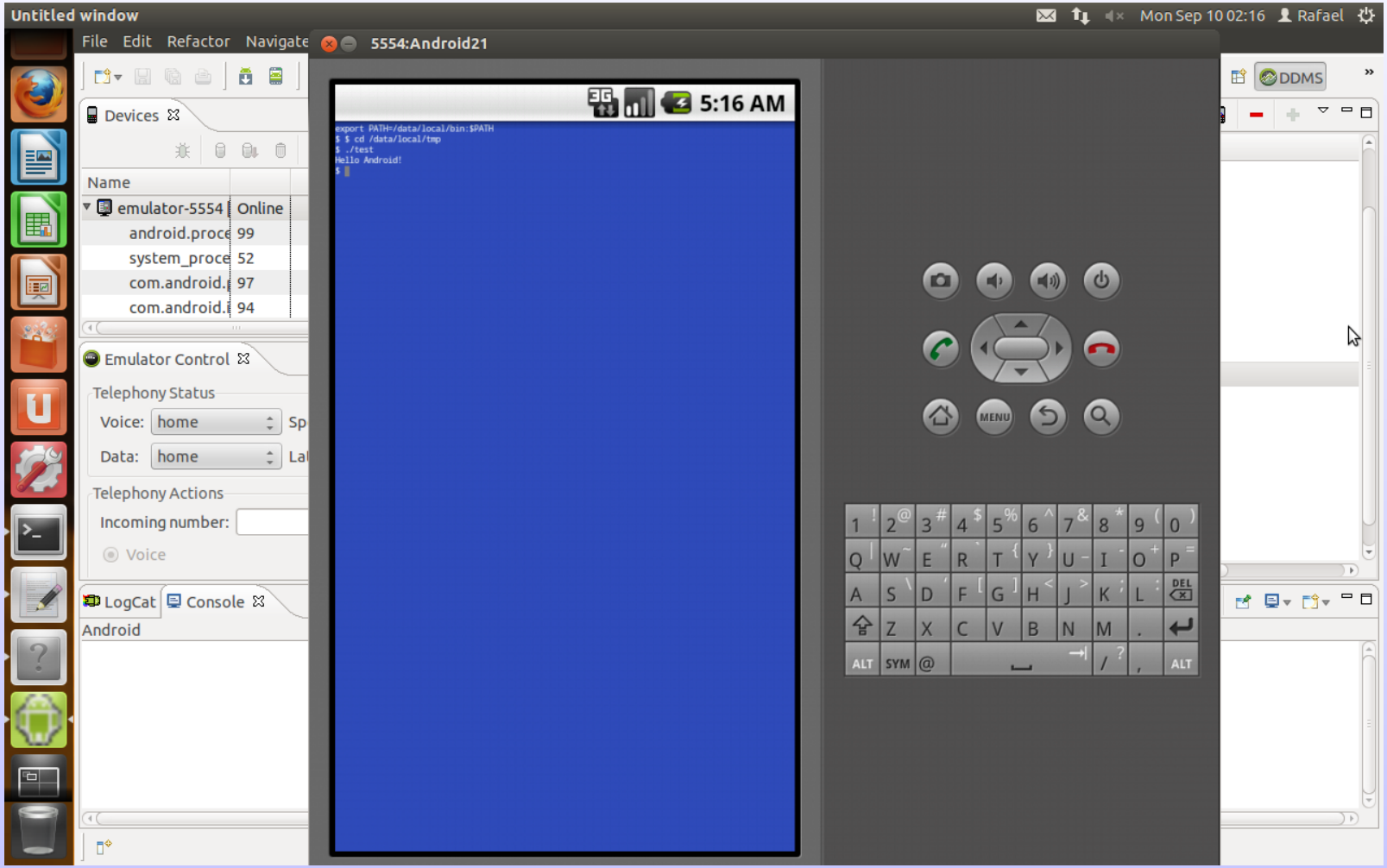


Figura: *Android Virtual Device Manager*

ADB - *Android Debug Bridge*

- Localizado no diretório: <sdk>/platform-tools/

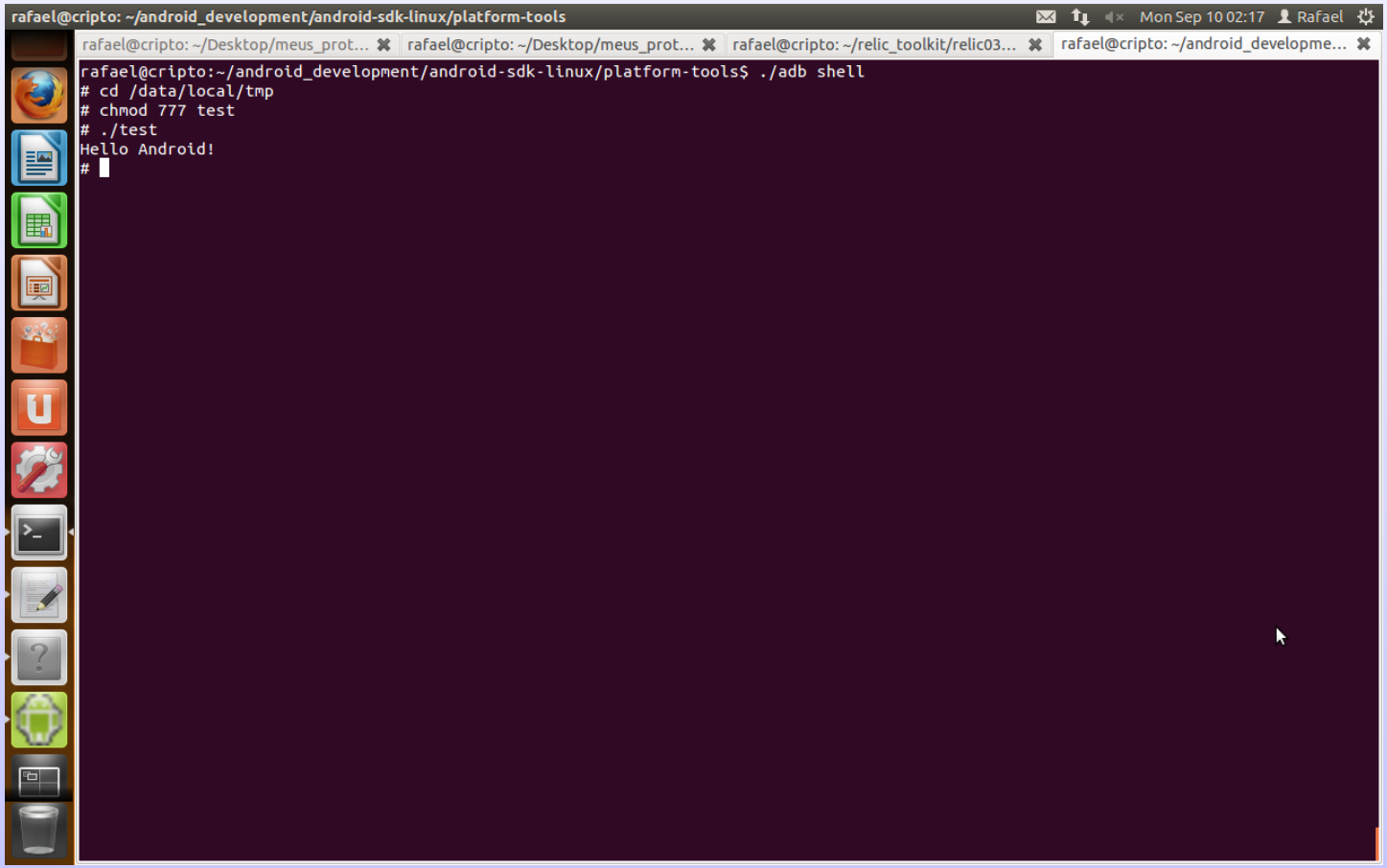


Figura: *Android Debug Bridge*

Relic-toolkit

- "RELIC is a modern cryptographic meta-toolkit with emphasis on efficiency and flexibility. RELIC can be used to build efficient and usable cryptographic toolkits tailored for specific security levels and algorithmic choices."
- Disponível em: http://code.google.com/p/relic-toolkit/

Relic-toolkit

Figura: Documentação da *Relic-toolkit*

Relic-toolkit

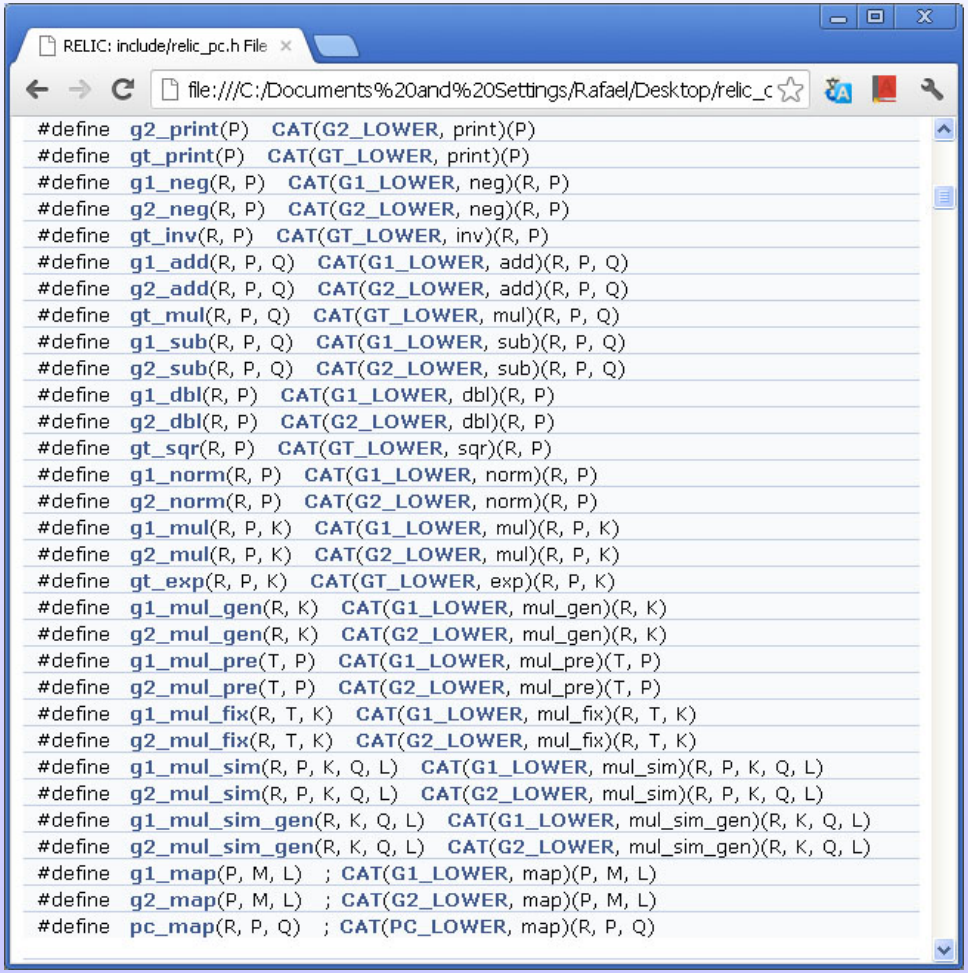


Figura: Documentação da *Relic-toolkit*

Relic-toolkit

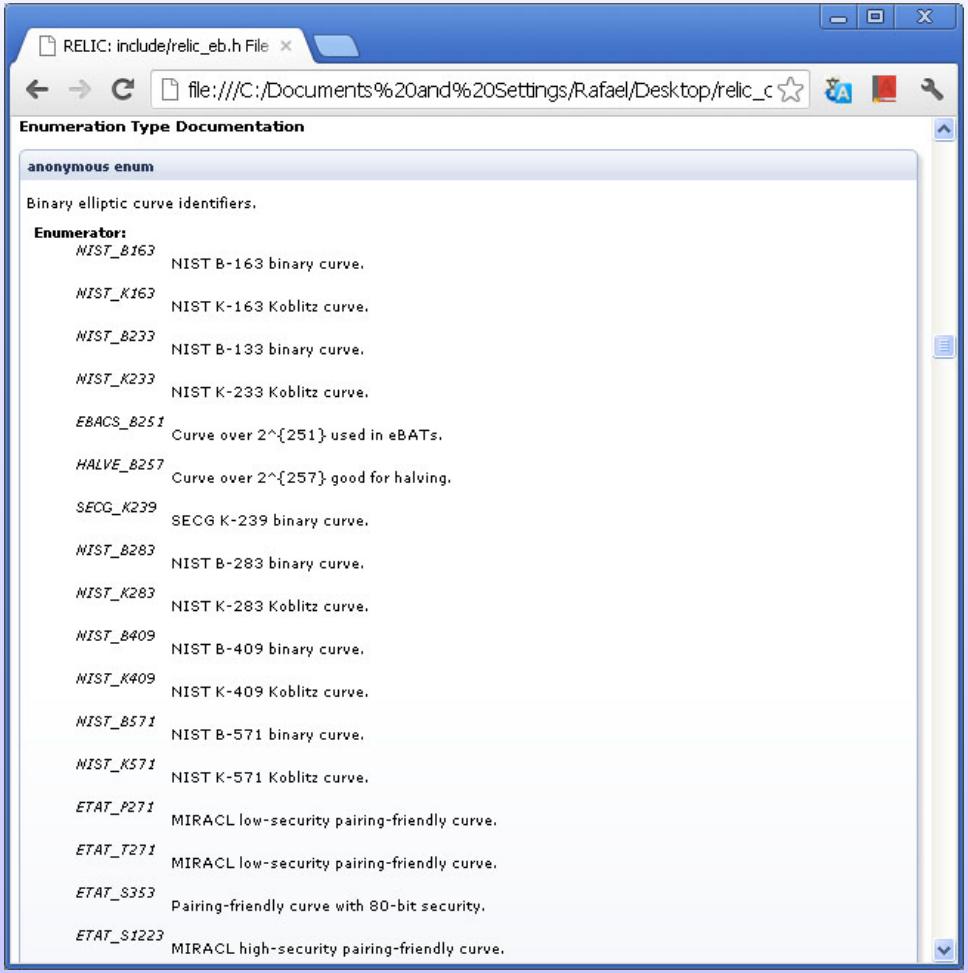


Figura: Curvas elípticas disponíveis para corpos binários

Relic-toolkit

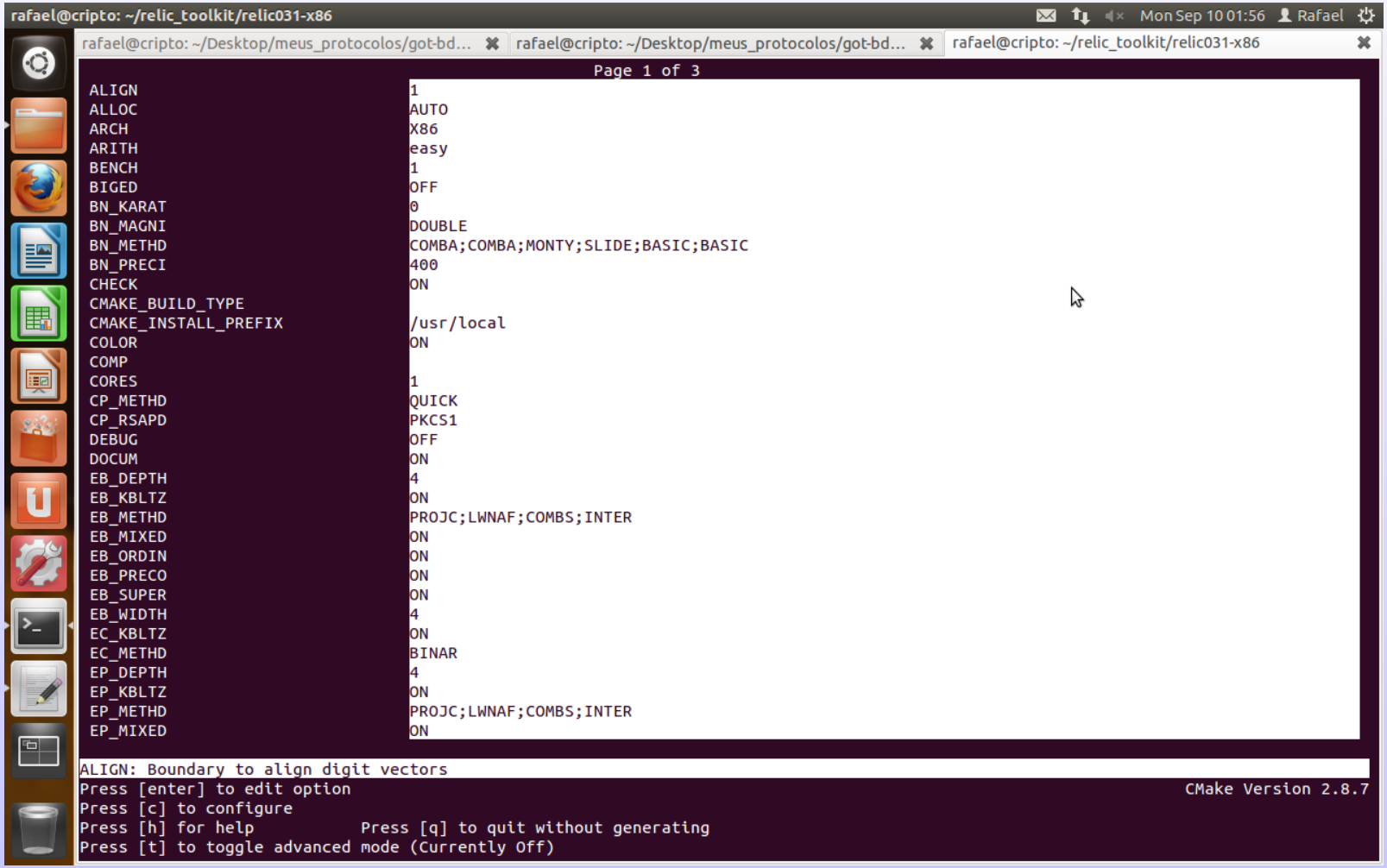
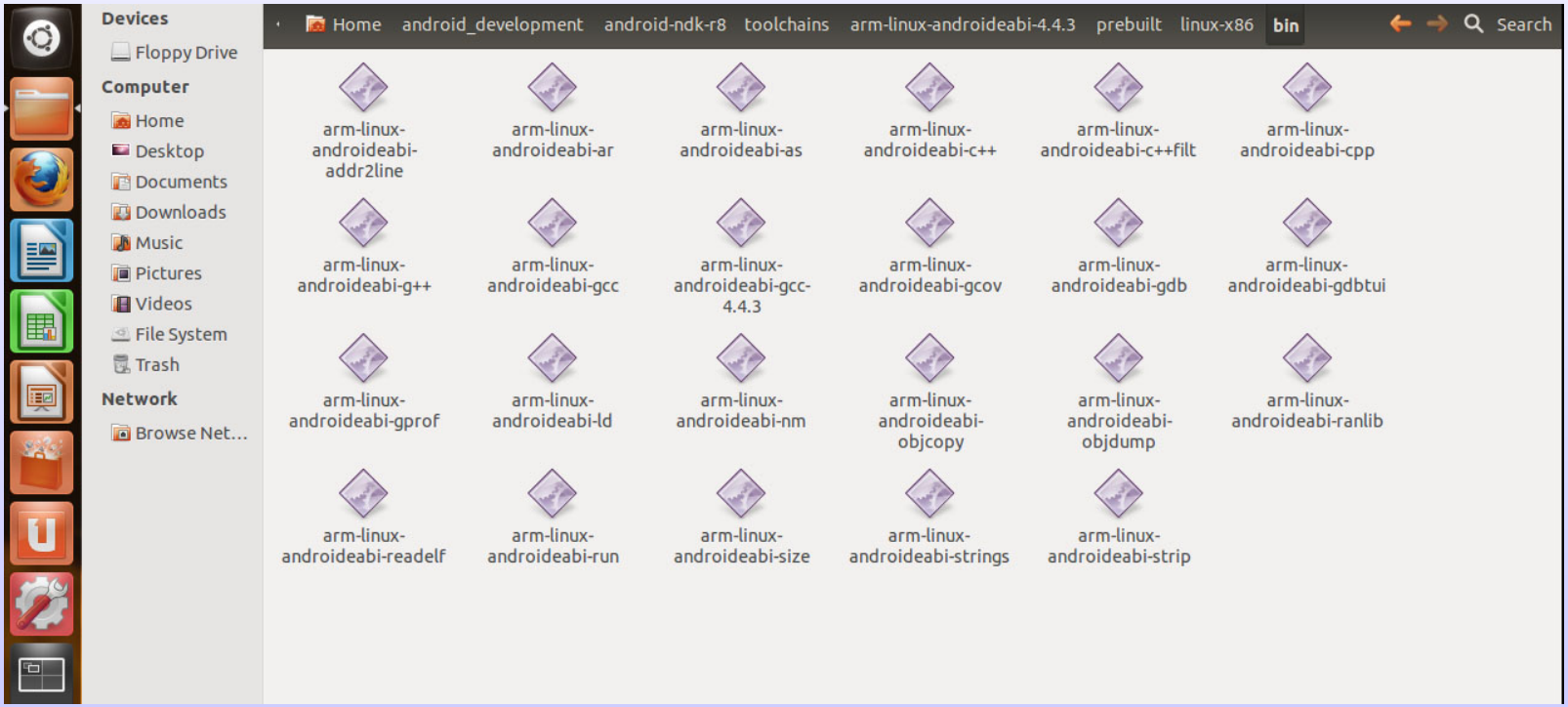


Figura: Algumas opções de compilação da *Relic-toolkit*

Android NDK Toolchains



Variáveis de ambiente

Open Save Undo

android_env.sh

```
1 #!/bin/sh
2 export NDK=$HOME/android_development/android-ndk-r8
3 SYSROOT=$NDK/platforms/android-9/arch-arm
4
5 MIDDLE=toolchains/arm-linux-androideabi-4.4.3/prebuilt/linux-x86/bin
6 PREF=arm-linux-androideabi-
7
8 export CC="$NDK/$MIDDLE/${PREF}gcc  --sysroot=$SYSROOT"
9 export CXX="$NDK/$MIDDLE/${PREF}g++  --sysroot=$SYSROOT"
10 export LD="$NDK/$MIDDLE/${PREF}ld  --sysroot=$SYSROOT"
11 export CPP="$NDK/$MIDDLE/${PREF}cpp  --sysroot=$SYSROOT"
12 export AS="$NDK/$MIDDLE/${PREF}as  --sysroot=$SYSROOT"
13 export OBJCOPY="$NDK/$MIDDLE/${PREF}objcopy  --sysroot=$SYSROOT"
14 export OBJDUMP="$NDK/$MIDDLE/${PREF}objdump  --sysroot=$SYSROOT"
15 export STRIP="$NDK/$MIDDLE/${PREF}strip  --sysroot=$SYSROOT"
16 export RANLIB="$NDK/$MIDDLE/${PREF}ranlib  --sysroot=$SYSROOT"
17 export CCLD="$NDK/$MIDDLE/${PREF}gcc  --sysroot=$SYSROOT"
18 export AR="$NDK/$MIDDLE/${PREF}ar  --sysroot=$SYSROOT"
```

Figura: Configurando algumas variáveis de ambiente para facilitar o processo de compilação

Ponto de Partida

- Projeto Borboleta (Fapesp).
  - Um dispositivo móvel (PDA) deseja se comunicar com um servidor de banco de dados de forma segura.
- Comparar o cenário existente no projeto Borboleta caso fossem utilizados outros protocolos de acordo de chave.
- Utilização da biblioteca **Relic Toolkit**, em linguagem C.
  - Suporte para cuvas elípticas.
  - Suporte para emparelhamentos bilineares.
  - Crescente uso pela comunidade acadêmica.
  - Documentação via *doxygen*, e frequente correção de *bugs*.

Cenário

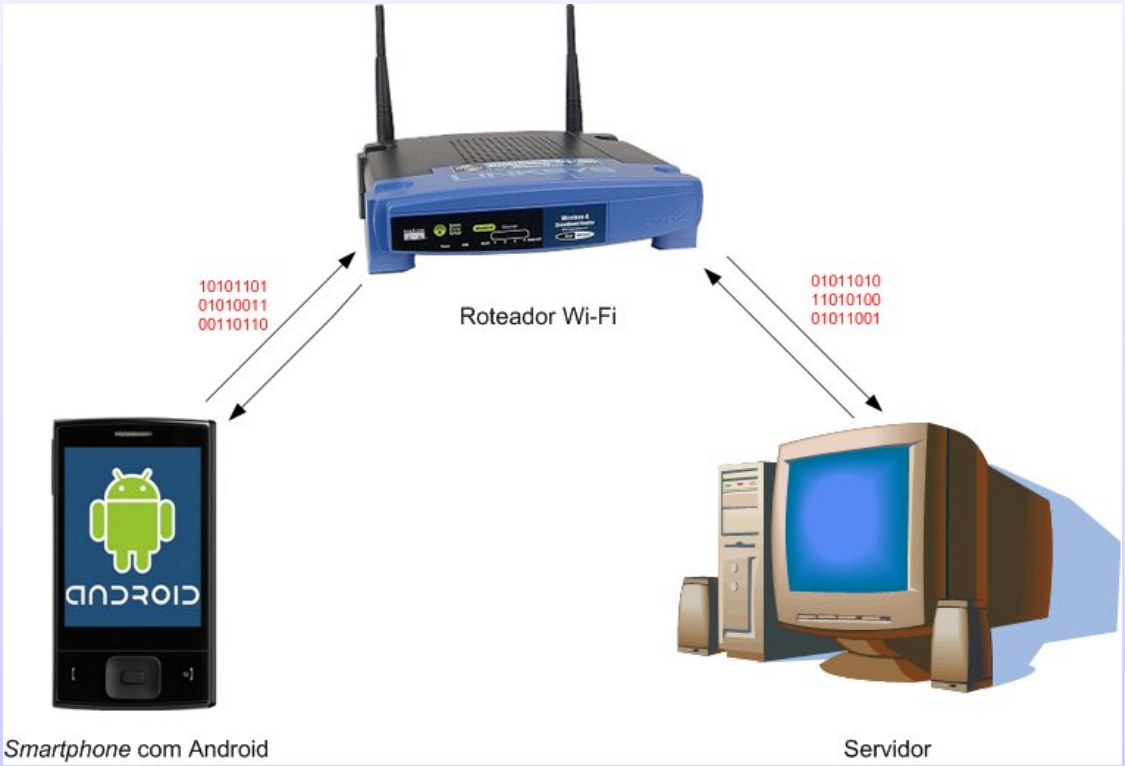


Figura: Cenário a ser estudado

Comunicação Servidor-*Smartphone*

- Escolhemos utilizar *sockets* em linguagem C.
  - Comunicação TCP/IP.
  - Integra-se facilmente aos outros códigos construídos (todo o código fica escrito em linguagem C).
  - Redução de *overhead*. Protocolos de comunicação mais complexos carregam outros "dados extras".
  - Ausência de funções na *Relic-toolkit* para transformação de certos tipos em *strings*.

Protocolos implementados

- Até o momento, implementamos 3 protocolos do modelo sem certificado:
  - LBG (BDH e Gap-BDH).
  - GOT (BDH e Gap-BDH).
  - GNT (Gap-BDH).
- Curvas utilizadas:
  - ETAT_T271 (segurança de 64 bits).
  - ETAT_S353 (segurança de 80 bits).
  - ETAT_S1223 (segurança de 128 bits).
- Versão da *Relic-toolkit*: 0.3.1

Equipamento utilizado

- Servidor:
  - Modelo: Compaq CQ-50 113BR.
  - Processador: Intel Core 2 Duo T5800, 2.0 GHz.
  - Memória RAM: 3GB DDR2, 800 MHz.
  - Conexão de rede: por cabo, 100 Mbps.
  - Sistema operacional: Ubuntu 10.04 LTS, 64 bits.
- *Smartphone*:
  - Modelo: Motorola Milestone.
  - Processador: ARM Cortex-A8, 600 MHz.
  - Memória RAM: 256 MB.
  - Conexão de rede: Wi-Fi (802.11g), 54 Mbps.
  - Sistema operacional: Android 2.2

Equipamento utilizado

- Roteador Wi-Fi:
  - Modelo: D-Link DIR-300.
  - Processador: Atheros AR2317, 183 MHz.
  - Memória RAM: 16 MB.
  - Conexão de rede: 4x 100 Mbps (cabo), e Wi-Fi de 54 Mbps.
  - Sistema operacional: DD-WRT v24.

IME - Instituto de Matemática e Estatística

Resultados (curva ETAT_T271)

	GNT (Gap-BDH)	GOT (Gap-BDH)	LBG (Gap-BDH)	GOT (BDH)	LGB (BDH)
Tempo total (10 rodadas) sem pré-computação	0,882	0,904	1,072	1,383	1,725
Tempo médio de uma única sessão	0,088	0,090	0,107	0,138	0,172
Tempo total (10 rodadas) com pré-computação	0,587	0,466	0,634	0,559	0,906
Tempo médio de uma única sessão	0,059	0,047	0,063	0,056	0,091
Memória utilizada pelas variáveis do programa (bytes)	2700	2364	2364	3132	3132
Tráfego na rede por sessão (bytes)	448	448	448	448	448

(Tempo em segundos)

Resultados (curva ETAT_T353)

	GNT (Gap-BDH)	GOT (Gap-BDH)	LBG (Gap-BDH)	GOT (BDH)	LGB (BDH)
Tempo total (10 rodadas) sem pré-computação	1,556	1,664	1,953	2,617	3,339
Tempo médio de uma única sessão	0,156	0,166	0,195	0,262	0,334
Tempo total (10 rodadas) com pré-computação	0,987	0,815	1,104	1,019	1,728
Tempo médio de uma única sessão	0,099	0,082	0,110	0,102	0,173
Memória utilizada pelas variáveis do programa (bytes)	3552	3108	3108	4128	4128
Tráfego na rede por sessão (bytes)	592	592	592	592	592

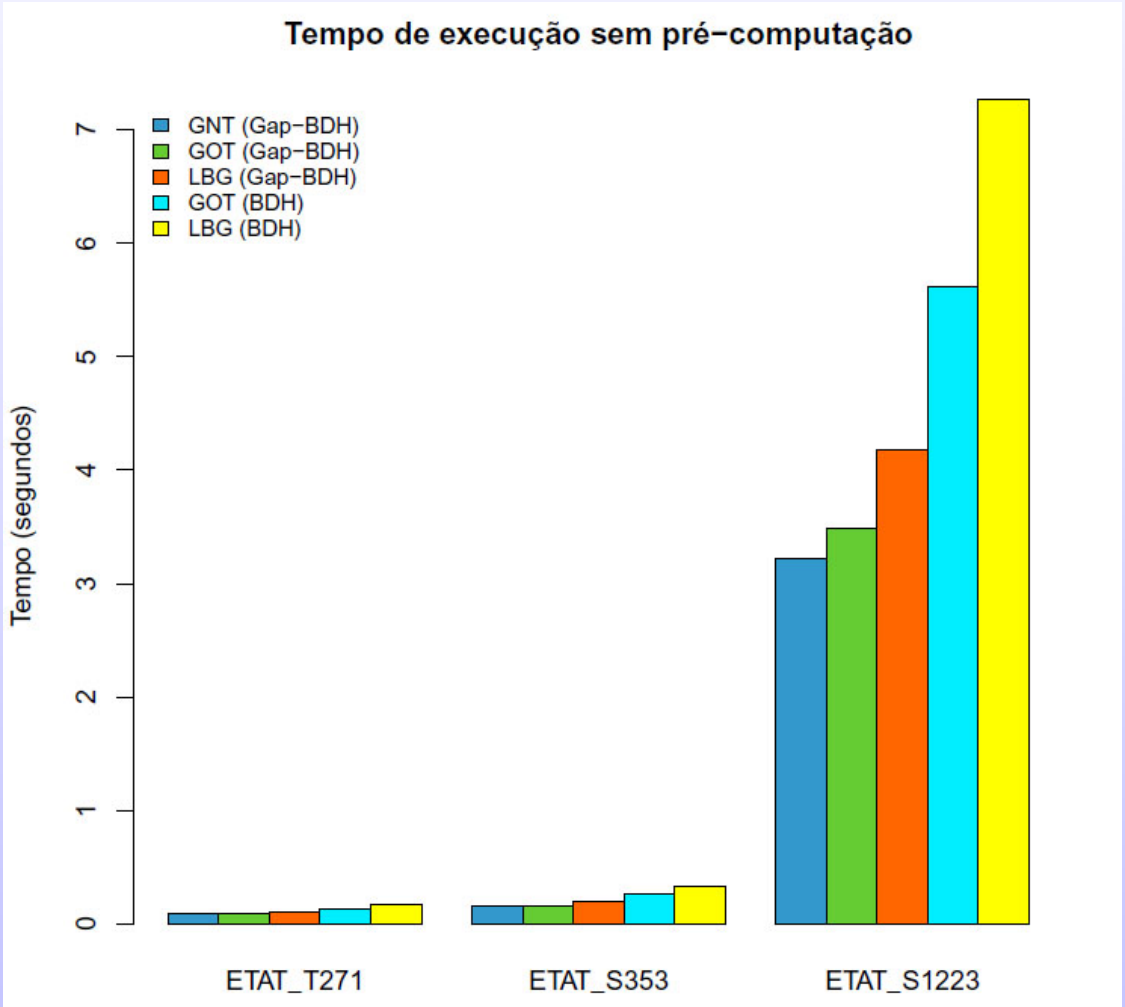
(Tempo em segundos)

Resultados (curva ETAT_T1223)

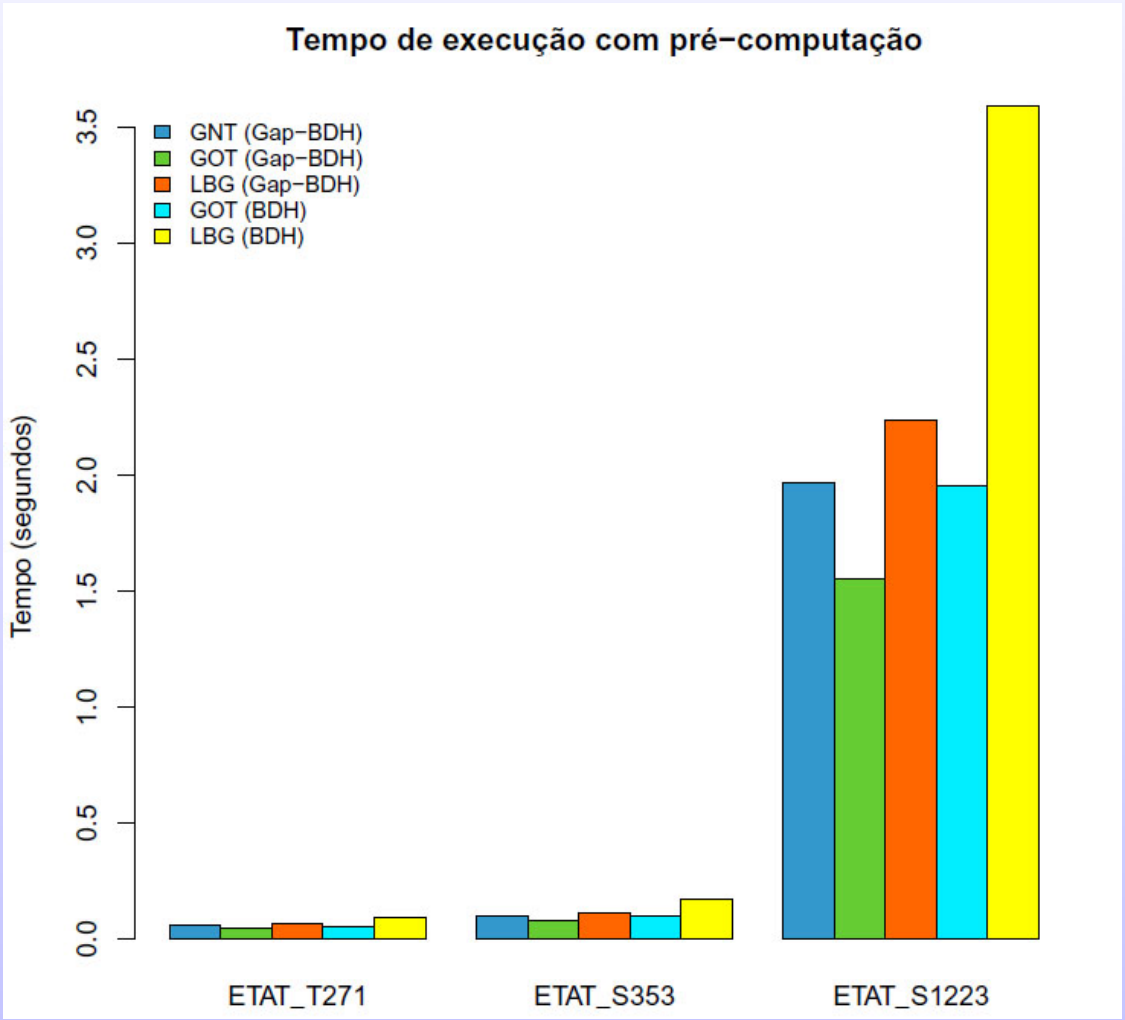
	GNT (Gap-BDH)	GOT (Gap-BDH)	LBG (Gap-BDH)	GOT (BDH)	LGB (BDH)
Tempo total (10 rodadas) sem pré-computação	32,169	34,852	41,833	56,134	72,629
Tempo médio de uma única sessão	3,217	3,485	4,183	5,613	7,263
Tempo total (10 rodadas) com pré-computação	19,642	15,503	22,368	19,554	35,885
Tempo médio de uma única sessão	1,964	1,550	2,237	1,955	3,588
Memória utilizada pelas variáveis do programa (bytes)	11244	9828	9828	13116	13116
Tráfego na rede por sessão (bytes)	1888	1888	1888	1888	1888

(Tempo em segundos)

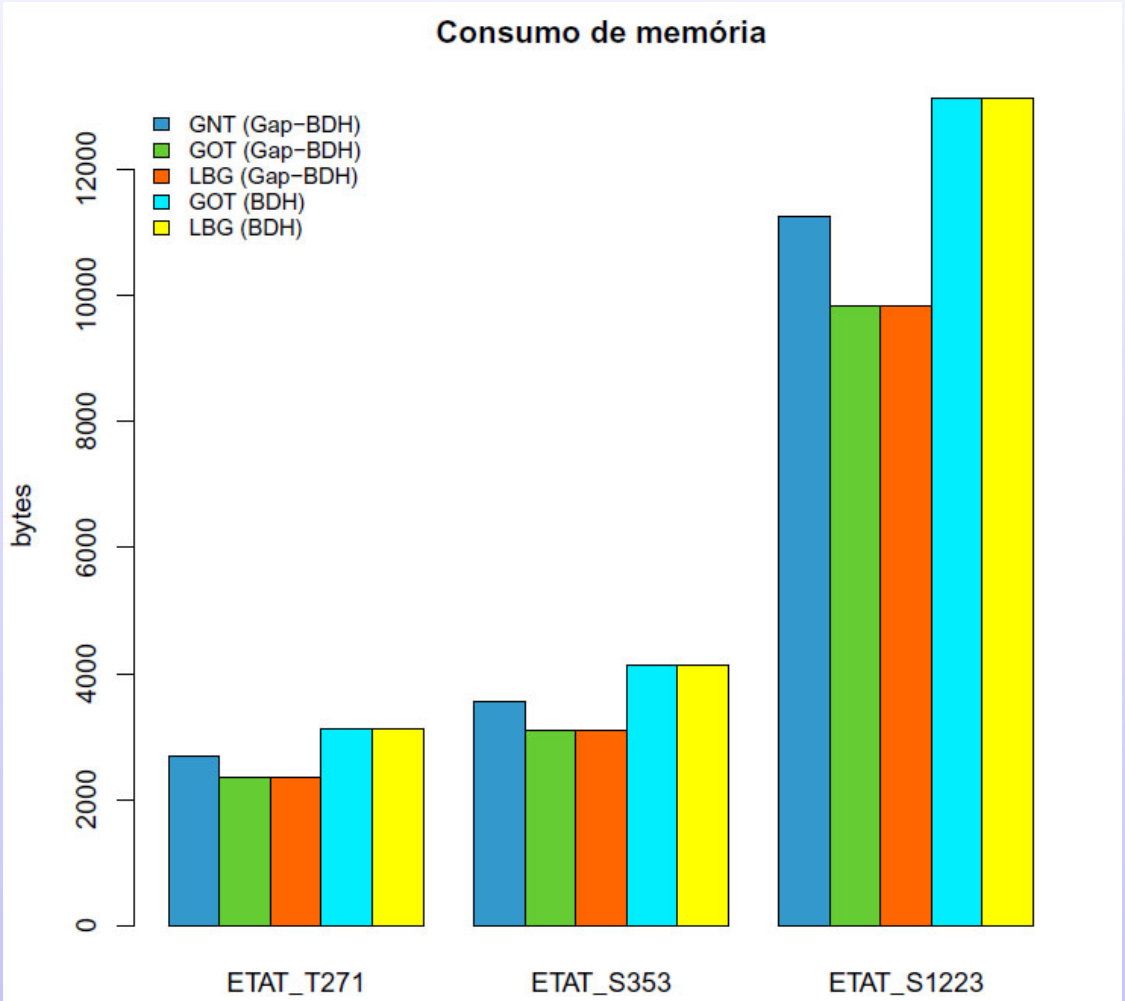
Comparação entre os protocolos (tempos sem pré-computação)



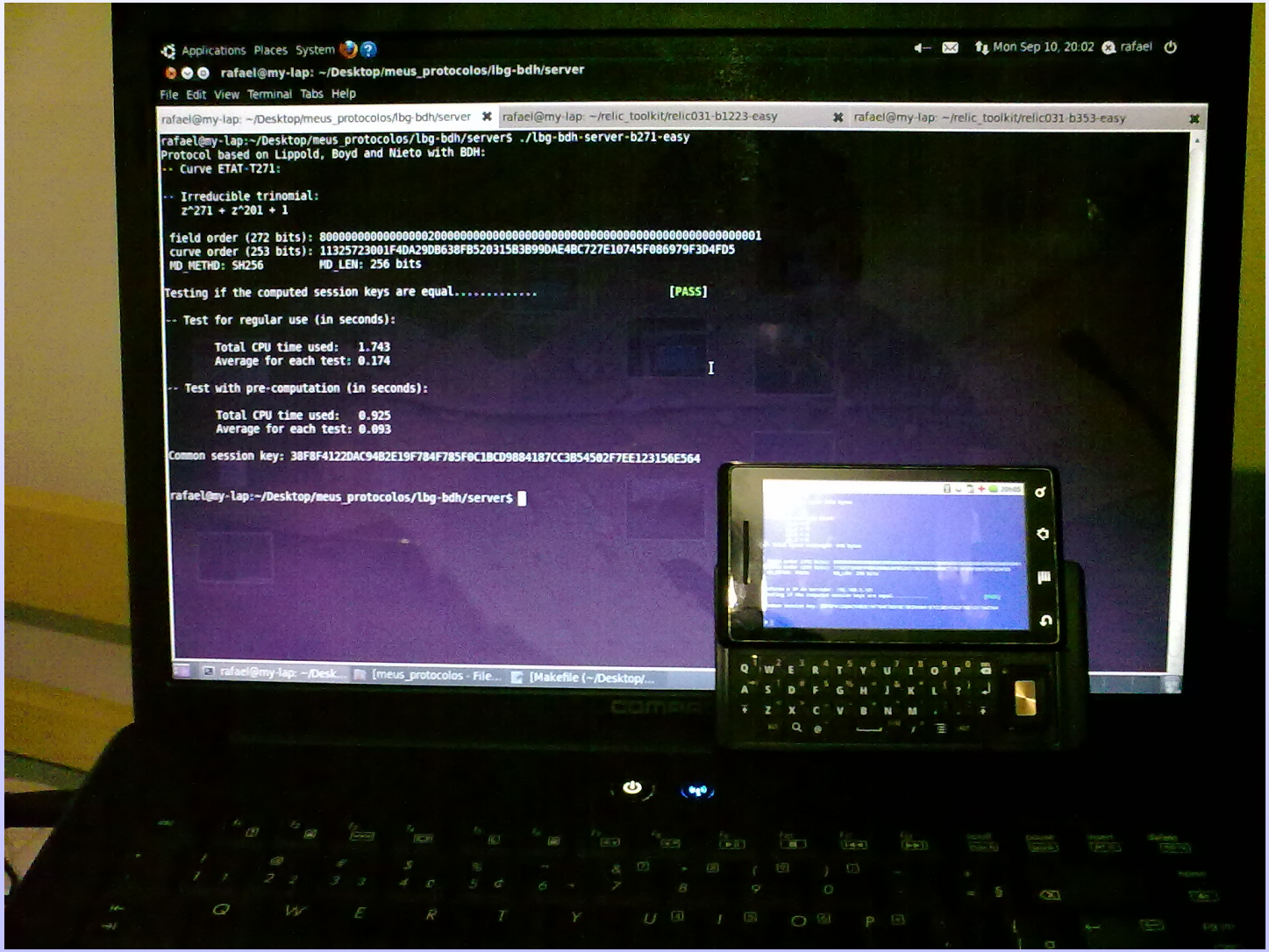
Comparação entre os protocolos (tempos com pré-computação)



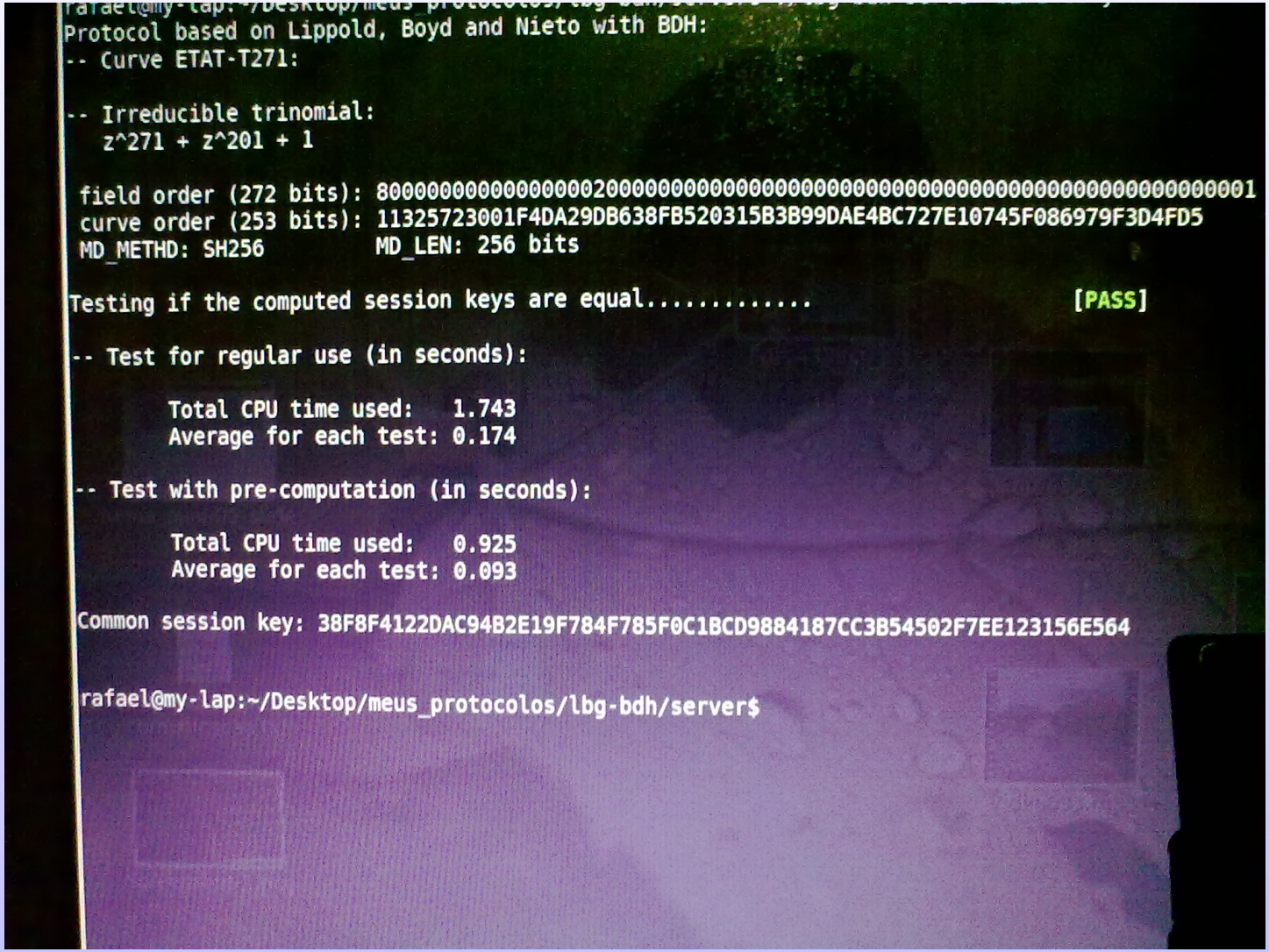
Comparação entre os protocolos (consumo de memória)



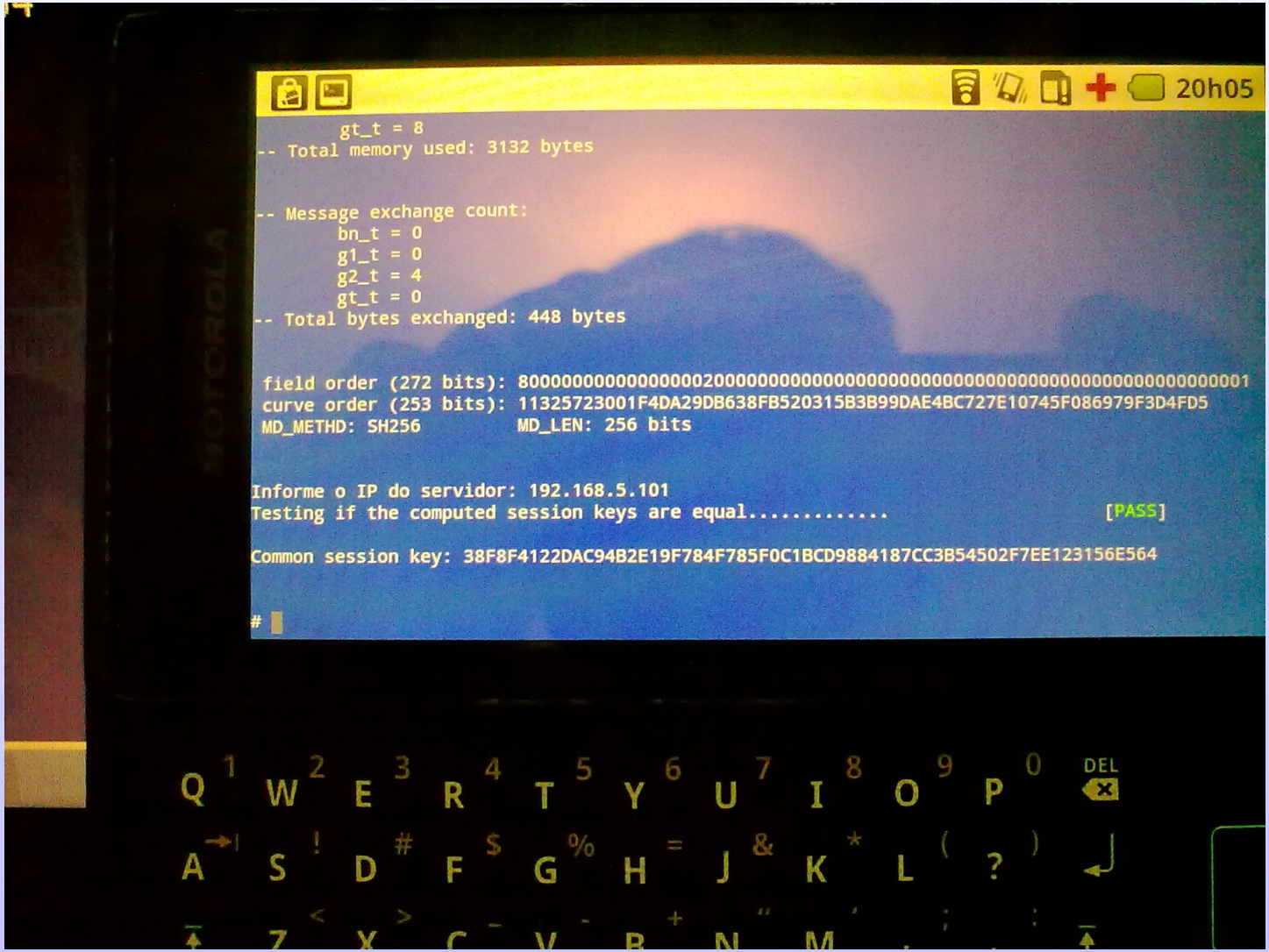
Algumas fotos extras...



Algumas fotos extras...



Algumas fotos extras...



Próximos passos

- Elaboração do relatório do CNPq.
- Escolha de novos protocolos para implementação.

Perguntas?