O Diabo Está nos Detalhes: uma Análise da Quebra da Assinatura Digital do PlayStation 3

Conferência CAOS 2018

Prof. Luis Menasché Schechter
luisms@dcc.ufrj.br

Universidade Federal do Rio de Janeiro

5 de Outubro de 2018

Introdução

- Nesta apresentação, apresentamos uma falha em um detalhe da implementação do esquema de assinatura digital do videogame PlayStation 3.
- Apesar da falha estar em um pequeno detalhe da implementação, ela permitiu a quebra completa do esquema de assinatura digital.
- Sendo assim, esta apresentação tem também o objetivo de alertar, através da apresentação de um caso concreto, um ponto importante em segurança da informação: muitos problemas de segurança são causados não por falhas nas especificações dos protocolos utilizados, mas por pequenos desvios (involuntários) em alguns pontos de sua implementação.

Visão Geral

- Em dezembro de 2010, o esquema de assinatura digital dos aplicativos (jogos, em sua maioria) que rodam no PlayStation 3 da Sony foi quebrado por um grupo de hackers europeus chamado *fail0verflow*.
- O grupo apresentou a quebra no congresso "27th Chaos Communication Congress" em Berlim, em 29/12/2010.
- Após a exibição, ficou claro que a quebra foi resultado muito mais de um erro em um detalhe da implementação por parte da Sony do que de qualquer habilidade extraordinária por parte dos hackers.

Visão Geral (2)

- O algoritmo de assinatura digital utilizado pela Sony é o DSA baseado em Curvas Elípticas.
- O DSA é um algoritmo de assinatura digital baseado em grupos, aparentado do algoritmo El Gamal.
- Neste caso particular, o grupo utilizado é um grupo obtido a partir de uma Curva Elíptica.
- O erro cometido pela Sony não surge de uma implementação errada da parte relativa às Curvas Elípticas.
- Ele também não surge de uma implementação errada do algoritmo DSA.
- Este é um erro muito mais fundamental que afeta qualquer esquema de assinatura baseado no El Gamal, seja qual for o grupo utilizado.

Hipótese Simplificadora

- Vamos descartar o uso de Curvas Elípticas e do algoritmo DSA e exibir o erro de implementação cometido pela Sony através do estudo do algoritmo de assinatura digital El Gamal tradicional.

Criptografia e Assinatura Digital

- Criptografia:
  - Alice quer enviar uma mensagem criptografada para Bernardo.
  - Bernardo possui uma chave pública de codificação e uma chave privada de decodificação.
  - Qualquer um (incluindo Alice) pode utilizar a chave pública de Bernardo para codificar mensagens destinadas a ele antes de enviá-las.
  - Apenas Bernardo pode decodificar (com sua chave privada) tais mensagens e acessar seu conteúdo original.
- Assinatura Digital:
  - Alice quer enviar uma mensagem assinada para Bernardo.
  - Alice possui uma chave privada de assinatura e uma chave pública de verificação.
  - Qualquer um (incluindo Bernardo) pode utilizar a chave pública de Alice para verificar a assinatura ao receber mensagens enviadas por ela.
  - Apenas Alice pode assinar corretamente (com sua chave privada) as mensagens.

Conceitos Básicos da Assinatura Digital El Gamal

- Desenvolvida pelo egípcio Taher El Gamal.
- É um método baseado no uso de grupos finitos cíclicos.
- Na versão tradicional do algoritmo, utiliza-se o conjunto

$$U(p) = \{1, 2, 3, \ldots, p-1\},$$

sendo p primo, com operações de produto e potenciação módulo p.

- Uma operação módulo p reduz os resultados maiores do que $p-1$ ao resto da divisão deste resultado por p.

Conceitos Básicos da Assinatura Digital El Gamal (2)

- Pelo *Teorema da Raiz Primitiva*, como p é primo, existe pelo menos um elemento $g \in U(p)$ tal que o conjunto $U(p)$ pode ser escrito como

$$U(p) = \{1, g, g^2, g^3, \ldots, g^{p-2}\},$$

  isto é, para qualquer elemento $h \in U(p)$, existe um expoente x no intervalo $0 \leq x \leq p-2$ tal que $h \equiv g^x \pmod{p}$.
- g é chamado de *gerador* do grupo cíclico $U(p)$.
- O *Problema do Logaritmo Discreto* consiste justamente em, dados g, h e p, determinar o valor do expoente x no intervalo acima tal que $g^x \equiv h \pmod{p}$.
- A resolução deste problema possui alta complexidade computacional no conjunto $U(p)$, quando p é grande.
- A segurança da assinatura El Gamal está baseada na dificuldade de resolução deste problema.

Variantes da Assinatura Digital El Gamal

- Possível Variante #1: utilizar outros grupos, que não sejam da forma $U(p)$.
  - Uma alternativa popular é a utilização de grupos construídos a partir de Curvas Elípticas.
- Possível Variante #2: utilizar estruturas de grupos mais complexas.
  - O método DSA realiza cálculos utilizando um par de grupos relacionados entre si.

Procedimentos da Assinatura El Gamal

- Geração do par de chaves:
  1. Seja p um primo e g um gerador do grupo cíclico $U(p)$.
  2. Selecione um número aleatório $1 < a < p - 1$.
  3. Calcule $y \equiv g^a$ em $U(p)$ (y é a forma reduzida de g^a módulo p).
  4. Os parâmetros públicos da implementação são p e g.
  5. A chave pública de verificação é y.
  6. A chave privada de assinatura é a.
- Para obter a chave privada a partir da chave pública é necessário resolver o problema do logaritmo discreto.

Procedimentos da Assinatura El Gamal (2)

- Assinatura:
  1. Seja m uma mensagem binária (no alfabeto $\{0,1\}$) a ser assinada e a a chave privada de assinatura.
  2. Seja $h : \{0,1\}^* \to \mathbb{Z}_p$ uma *função de hash*, que também é um parâmetro público da implementação.
  3. Selecione um número aleatório $1 \leq k \leq p-2$ tal que $\mathrm{mdc}(k, p-1) = 1$.
  4. Calcule $r \equiv g^k \pmod{p}$.
  5. Calcule o inverso de k módulo $p-1$ (denotado por k^{-1}).
  6. Calcule $s \equiv k^{-1} * (h(m) - ar) \pmod{p-1}$.
  7. A assinatura para m é o par (r, s).

Procedimentos da Assinatura El Gamal (3)

- Verificação:
  1. Seja y a chave de verificação, m a mensagem e (r, s) a assinatura a ser verificada.
  2. Verifique que $1 \leq r \leq p - 1$. Senão, rejeite a assinatura.
  3. Calcule $v_1 \equiv y^r * r^s \pmod{p}$.
  4. Calcule $h(m)$ e $v_2 \equiv g^{h(m)} \pmod{p}$.
  5. Aceite a assinatura se e somente se $v_1 = v_2$.

Por que o processo de verificação funciona?

Inicialmente,

v_1

Por que o processo de verificação funciona?

Inicialmente,

$$v_1 \equiv y^r * r^s$$

Por que o processo de verificação funciona?

Inicialmente,

$$v_1 \equiv y^r * r^s \equiv (g^a)^r * r^s$$

Por que o processo de verificação funciona?

Inicialmente,

$$v_1 \equiv y^r * r^s \equiv (g^a)^r * r^s \equiv g^{ar} * r^s$$

Por que o processo de verificação funciona?

Inicialmente,

$$v_1 \equiv y^r * r^s \equiv (g^a)^r * r^s \equiv g^{ar} * r^s \equiv g^{ar} * (g^k)^s$$

Por que o processo de verificação funciona?

Inicialmente,

$$v_1 \equiv y^r * r^s \equiv (g^a)^r * r^s \equiv g^{ar} * r^s \equiv g^{ar} * (g^k)^s \equiv g^{ar} * g^{ks}$$

Por que o processo de verificação funciona?

Inicialmente,

$$v_1 \equiv y^r * r^s \equiv (g^a)^r * r^s \equiv g^{ar} * r^s \equiv g^{ar} * (g^k)^s \equiv g^{ar} * g^{ks} \equiv$$

$$\equiv g^{ar+ks} \pmod{p}$$

Por que o processo de verificação funciona?

Inicialmente,

$$v_1 \equiv y^r * r^s \equiv (g^a)^r * r^s \equiv g^{ar} * r^s \equiv g^{ar} * (g^k)^s \equiv g^{ar} * g^{ks} \equiv$$

$$\equiv g^{ar+ks} \pmod{p}$$

Temos

$$s \equiv k^{-1} * (h(m) - ar) \pmod{p-1}.$$

Por que o processo de verificação funciona?

Inicialmente,

$$v_1 \equiv y^r * r^s \equiv (g^a)^r * r^s \equiv g^{ar} * r^s \equiv g^{ar} * (g^k)^s \equiv g^{ar} * g^{ks} \equiv$$

$$\equiv g^{ar+ks} \pmod{p}$$

Temos

$$s \equiv k^{-1} * (h(m) - ar) \pmod{p-1}.$$

Logo,

$$ks \equiv (h(m) - ar) \pmod{p-1},$$

Por que o processo de verificação funciona?

Inicialmente,

$$v_1 \equiv y^r * r^s \equiv (g^a)^r * r^s \equiv g^{ar} * r^s \equiv g^{ar} * (g^k)^s \equiv g^{ar} * g^{ks} \equiv$$

$$\equiv g^{ar+ks} \pmod{p}$$

Temos

$$s \equiv k^{-1} * (h(m) - ar) \pmod{p-1}.$$

Logo,

$$ks \equiv (h(m) - ar) \pmod{p-1},$$

o que implica

$$ks + ar \equiv h(m) \pmod{p-1}.$$

Por que o processo de verificação funciona?

Inicialmente,

$$v_1 \equiv y^r * r^s \equiv (g^a)^r * r^s \equiv g^{ar} * r^s \equiv g^{ar} * (g^k)^s \equiv g^{ar} * g^{ks} \equiv$$

$$\equiv g^{ar+ks} \pmod{p}$$

Temos

$$s \equiv k^{-1} * (h(m) - ar) \pmod{p-1}.$$

Logo,

$$ks \equiv (h(m) - ar) \pmod{p-1},$$

o que implica

$$ks + ar \equiv h(m) \pmod{p-1}.$$

Sem a congruência, temos

$$ks + ar = h(m) + (p-1) * l.$$

Por que o processo de verificação funciona? (2)

Assim,

$$v_1$$

Por que o processo de verificação funciona? (2)

Assim,

$$v_1 \equiv g^{ar+ks}$$

Por que o processo de verificação funciona? (2)

Assim,

$$v_1 \equiv g^{ar+ks} \equiv g^{h(m)+(p-1)*l}$$

Por que o processo de verificação funciona? (2)

Assim,

$$v_1 \equiv g^{ar+ks} \equiv g^{h(m)+(p-1)*l} \equiv g^{h(m)} * g^{(p-1)*l}$$

Por que o processo de verificação funciona? (2)

Assim,

$$v_1 \equiv g^{ar+ks} \equiv g^{h(m)+(p-1)*l} \equiv g^{h(m)} * g^{(p-1)*l} \equiv$$

$$\equiv g^{h(m)} * (g^{p-1})^{l} \pmod{p}$$

Por que o processo de verificação funciona? (2)

Assim,

$$v_1 \equiv g^{ar+ks} \equiv g^{h(m)+(p-1)*l} \equiv g^{h(m)} * g^{(p-1)*l} \equiv$$

$$\equiv g^{h(m)} * (g^{p-1})^l \pmod{p}$$

Devido a um resultado conhecido como *Pequeno Teorema de Fermat*, temos

$$g^{p-1} \equiv 1 \pmod{p}.$$

Por que o processo de verificação funciona? (2)

Assim,

$$v_1 \equiv g^{ar+ks} \equiv g^{h(m)+(p-1)*l} \equiv g^{h(m)} * g^{(p-1)*l} \equiv$$

$$\equiv g^{h(m)} * (g^{p-1})^l \pmod{p}$$

Devido a um resultado conhecido como *Pequeno Teorema de Fermat*, temos

$$g^{p-1} \equiv 1 \pmod{p}.$$

Assim,

v_1

Por que o processo de verificação funciona? (2)

Assim,

$$v_1 \equiv g^{ar+ks} \equiv g^{h(m)+(p-1)*l} \equiv g^{h(m)} * g^{(p-1)*l} \equiv$$

$$\equiv g^{h(m)} * (g^{p-1})^l \pmod{p}$$

Devido a um resultado conhecido como *Pequeno Teorema de Fermat*, temos

$$g^{p-1} \equiv 1 \pmod{p}.$$

Assim,

$$v_1 \equiv g^{h(m)} * (g^{p-1})^l$$

Por que o processo de verificação funciona? (2)

Assim,

$$v_1 \equiv g^{ar+ks} \equiv g^{h(m)+(p-1)*l} \equiv g^{h(m)} * g^{(p-1)*l} \equiv$$

$$\equiv g^{h(m)} * (g^{p-1})^l \pmod{p}$$

Devido a um resultado conhecido como *Pequeno Teorema de Fermat*, temos

$$g^{p-1} \equiv 1 \pmod{p}.$$

Assim,

$$v_1 \equiv g^{h(m)} * (g^{p-1})^l \equiv g^{h(m)} * (1)^l$$

Por que o processo de verificação funciona? (2)

Assim,

$$v_1 \equiv g^{ar+ks} \equiv g^{h(m)+(p-1)*l} \equiv g^{h(m)} * g^{(p-1)*l} \equiv$$

$$\equiv g^{h(m)} * (g^{p-1})^l \pmod{p}$$

Devido a um resultado conhecido como *Pequeno Teorema de Fermat*, temos

$$g^{p-1} \equiv 1 \pmod{p}.$$

Assim,

$$v_1 \equiv g^{h(m)} * (g^{p-1})^l \equiv g^{h(m)} * (1)^l \equiv g^{h(m)}$$

Por que o processo de verificação funciona? (2)

Assim,

$$v_1 \equiv g^{ar+ks} \equiv g^{h(m)+(p-1)*l} \equiv g^{h(m)} * g^{(p-1)*l} \equiv$$

$$\equiv g^{h(m)} * (g^{p-1})^l \pmod{p}$$

Devido a um resultado conhecido como *Pequeno Teorema de Fermat*, temos

$$g^{p-1} \equiv 1 \pmod{p}.$$

Assim,

$$v_1 \equiv g^{h(m)} * (g^{p-1})^l \equiv g^{h(m)} * (1)^l \equiv g^{h(m)} \equiv v_2 \pmod{p}.$$

Segurança da Assinatura El Gamal

- É fundamental que, a cada nova mensagem que vá ser assinada, um novo valor aleatório de k seja gerado. Utilizar o mesmo k para diversas mensagens torna o sistema muito frágil.
- Caso nenhuma função hash seja utilizada, abre-se uma brecha para a geração de assinaturas falsas.
- Caso o passo 2 da verificação da assinatura não seja feito, abre-se outra brecha para a geração de assinaturas falsas.
- Finalmente, precisa-se escolher um primo p grande, pois com p pequeno a resolução do problema do logaritmo discreto torna-se realizável na prática e pode-se obter a chave privada diretamente da chave pública.

A Assinatura Digital no PlayStation

- No caso do PlayStation, não estamos interessados em assinar mensagens que irão trafegar em uma rede.
- Estamos interessados em assinar softwares produzidos para serem executados no console PlayStation.
- A Sony cria um par de chaves (chave de assinatura (privada) / chave de verificação (pública)) e grava a chave de verificação junto com o algoritmo de verificação de assinaturas no hardware ou no firmware do console.
- Todo software produzido para o PlayStation precisa ser "autorizado" pela Sony para que ele possa ser executado no console.
- Esta "autorização" consiste em anexar ao código do software uma assinatura válida para este código com a chave privada que é mantida em propriedade da Sony.

A Assinatura Digital no PlayStation (2)

- Quando qualquer software é carregado no console, o console executa sobre o código do software o algoritmo de verificação de assinatura.
- Caso a assinatura seja aceita, o software roda normalmente. Caso contrário, o software não é autorizado pela Sony e a execução é abortada.
- Para todos os efeitos da nossa discussão, podemos considerar que o algoritmo de assinatura digital utilizado no PlayStation 3 é o El Gamal.
- Caso outras pessoas ou entidades fora da Sony obtenham a chave privada de assinatura, elas podem assinar qualquer software livremente, de maneira que ele seja executado sem problemas no console.
- Isto permitiria a execução de software pirata, de software homebrew e até mesmo o uso de um console PlayStation como um computador Linux.

O Erro da Sony

O Erro da Sony

- **A Sony não utilizou valores aleatórios distintos para o parâmetro k em cada processo de assinatura. Ela utilizou sempre o mesmo valor de k.**

O Erro da Sony

- **A Sony não utilizou valores aleatórios distintos para o parâmetro k em cada processo de assinatura. Ela utilizou sempre o mesmo valor de k.**
- Handbook of Applied Cryptography: "A different k must be selected for each message signed; otherwise, the private key can be determined with high probability".

O Erro da Sony

- **A Sony não utilizou valores aleatórios distintos para o parâmetro k em cada processo de assinatura. Ela utilizou sempre o mesmo valor de k.**
- Handbook of Applied Cryptography: "A different k must be selected for each message signed; otherwise, the private key can be determined with high probability".
- Wikipedia (ElGamal Signature Scheme): "The signer must be careful to choose a different k uniformly at random for each signature and to be certain that k, or even partial information about k, is not leaked. Otherwise, an attacker may be able to deduce the secret key x with reduced difficulty, perhaps enough to allow a practical attack. In particular, if two messages are sent using the same value of k and the same key, then an attacker can compute x directly."

O Erro da Sony (2)

- "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms" (artigo original de T. El Gamal, **1985**): "If any k is used twice in the signing, then the system of equations is uniquely determined and x can be recovered. So for the system to be secure, any value of k should never be used twice."

O Erro da Sony (2)

- "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms" (artigo original de T. El Gamal, **1985**): "If any k is used twice in the signing, then the system of equations is uniquely determined and x can be recovered. So for the system to be secure, any value of k should never be used twice."
- Aparentemente, a Sony não sabia que utilizar o mesmo valor de k repetidas vezes geraria uma fragilidade no sistema.
- Vamos então mostrar como obter a chave privada de assinatura a se temos a assinatura de duas mensagens distintas que foram geradas com o mesmo parâmetro k.
- As contas envolvidas são extremamente elementares. Ou seja, esta falha permite a quebra do sistema em pouco tempo e sem nenhuma necessidade especial de processamento pesado.

Obtendo a Chave Privada de Assinatura

- Parâmetros públicos: h, g e p.

Obtendo a Chave Privada de Assinatura

- Parâmetros públicos: h, g e p.
- Parâmetro k igual para duas mensagens.

Obtendo a Chave Privada de Assinatura

- Parâmetros públicos: h, g e p.
- Parâmetro k igual para duas mensagens.
- Mensagens: m_1 e m_2.

Obtendo a Chave Privada de Assinatura

- Parâmetros públicos: h, g e p.
- Parâmetro k igual para duas mensagens.
- Mensagens: m_1 e m_2.
- Assinatura de m_1:

Obtendo a Chave Privada de Assinatura

- Parâmetros públicos: h, g e p.
- Parâmetro k igual para duas mensagens.
- Mensagens: m_1 e m_2.
- Assinatura de m_1:

$$r \equiv g^k \pmod{p},$$

Obtendo a Chave Privada de Assinatura

- Parâmetros públicos: h, g e p.
- Parâmetro k igual para duas mensagens.
- Mensagens: m_1 e m_2.
- Assinatura de m_1:

$$r \equiv g^k \pmod{p},$$

$$s_1 \equiv k^{-1}(h(m_1) - ar) \pmod{p-1}.$$

Obtendo a Chave Privada de Assinatura

- Parâmetros públicos: h, g e p.
- Parâmetro k igual para duas mensagens.
- Mensagens: m_1 e m_2.
- Assinatura de m_1:

$$r \equiv g^k \pmod{p},$$

$$s_1 \equiv k^{-1}(h(m_1) - ar) \pmod{p-1}.$$

- Assinatura de m_2:

Obtendo a Chave Privada de Assinatura

- Parâmetros públicos: h, g e p.
- Parâmetro k igual para duas mensagens.
- Mensagens: m_1 e m_2.
- Assinatura de m_1:

$$r \equiv g^k \pmod{p},$$

$$s_1 \equiv k^{-1}(h(m_1) - ar) \pmod{p-1}.$$

- Assinatura de m_2:

$$r \equiv g^k \pmod{p},$$

Obtendo a Chave Privada de Assinatura

- Parâmetros públicos: h, g e p.
- Parâmetro k igual para duas mensagens.
- Mensagens: m_1 e m_2.
- Assinatura de m_1:

$$r \equiv g^k \pmod{p},$$

$$s_1 \equiv k^{-1}(h(m_1) - ar) \pmod{p-1}.$$

- Assinatura de m_2:

$$r \equiv g^k \pmod{p},$$

$$s_2 \equiv k^{-1}(h(m_2) - ar) \pmod{p-1}.$$

Obtendo a Chave Privada de Assinatura (2)

Temos então

$$ks_1 \equiv h(m_1) - ar \pmod{p-1}$$

e

$$ks_2 \equiv h(m_2) - ar \pmod{p-1}.$$

Obtendo a Chave Privada de Assinatura (2)

Temos então

$$ks_1 \equiv h(m_1) - ar \pmod{p-1}$$

e

$$ks_2 \equiv h(m_2) - ar \pmod{p-1}.$$

Assim,

$$k(s_1 - s_2) \equiv h(m_1) - h(m_2) \pmod{p-1}.$$

Obtendo a Chave Privada de Assinatura (2)

Temos então

$$ks_1 \equiv h(m_1) - ar \pmod{p-1}$$

e

$$ks_2 \equiv h(m_2) - ar \pmod{p-1}.$$

Assim,

$$k(s_1 - s_2) \equiv h(m_1) - h(m_2) \pmod{p-1}.$$

Se $\mathrm{mdc}(s_1 - s_2, p-1) = 1$, podemos calcular $(s_1 - s_2)^{-1}$ e então temos

$$k \equiv (s_1 - s_2)^{-1}(h(m_1) - h(m_2)) \pmod{p-1}.$$

Obtendo a Chave Privada de Assinatura (3)

Com o valor de k encontrado, temos

$$ar \equiv h(m_i) - ks_i \pmod{p-1}, \text{ para } i = 1, 2.$$

Obtendo a Chave Privada de Assinatura (3)

Com o valor de k encontrado, temos

$$ar \equiv h(m_i) - ks_i \pmod{p-1}, \text{ para } i = 1, 2.$$

Se $\mathrm{mdc}(r, p-1) = 1$, podemos calcular r^{-1} e então temos o valor da chave privada

$$a \equiv r^{-1}(h(m_i) - ks_i) \pmod{p-1}.$$

Obtendo a Chave Privada de Assinatura (3)

Com o valor de k encontrado, temos

$$ar \equiv h(m_i) - ks_i \pmod{p-1}, \text{ para } i = 1, 2.$$

Se $\mathrm{mdc}(r, p-1) = 1$, podemos calcular r^{-1} e então temos o valor da chave privada

$$a \equiv r^{-1}(h(m_i) - ks_i) \pmod{p-1}.$$

Mesmo que $\mathrm{mdc}(r, p-1)$ seja diferente de 1 e não possamos obter o valor da chave privada, ainda temos informação suficiente para gerar assinaturas que serão aceitas pelo algoritmo de verificação.

Obtendo a Chave Privada de Assinatura (4)

- Como temos o valor de k e o parâmetro g é público, temos o valor de

$$r \equiv g^k \pmod{p}$$

e o valor de k^{-1}.

Obtendo a Chave Privada de Assinatura (4)

- Como temos o valor de k e o parâmetro g é público, temos o valor de

$$r \equiv g^k \pmod{p}$$

  e o valor de k^{-1}.
- Podemos também calcular o valor de

$$ar \equiv h(m_i) - ks_i \pmod{p-1}.$$

Obtendo a Chave Privada de Assinatura (4)

- Como temos o valor de k e o parâmetro g é público, temos o valor de

$$r \equiv g^k \pmod{p}$$

e o valor de k^{-1}.

- Podemos também calcular o valor de

$$ar \equiv h(m_i) - ks_i \pmod{p-1}.$$

- Se queremos assinar uma mensagem m^*, precisamos obter um valor s^* de forma que o par (r, s^*) seja uma assinatura válida para m^* de acordo com a chave privada de assinatura a (que não sabemos exatamente qual é).

Obtendo a Chave Privada de Assinatura (4)

- Como temos o valor de k e o parâmetro g é público, temos o valor de

$$r \equiv g^k \pmod{p}$$

e o valor de k^{-1}.

- Podemos também calcular o valor de

$$ar \equiv h(m_i) - ks_i \pmod{p-1}.$$

- Se queremos assinar uma mensagem m^*, precisamos obter um valor s^* de forma que o par (r, s^*) seja uma assinatura válida para m^* de acordo com a chave privada de assinatura a (que não sabemos exatamente qual é).
- Pela equação do algoritmo de geração da assinatura, temos que

$$s^* \equiv k^{-1}(h(m^*) - ar).$$

Obtendo a Chave Privada de Assinatura (5)

- Como conhecemos k^{-1}, h e o valor do produto ar, podemos calcular s^*.
- Temos então uma assinatura válida para a mensagem m^*.

Conclusões

- Um hacker que não é ligado ao grupo *fail0verflow* calculou, após a apresentação do congresso, a chave privada de assinatura do PlayStation 3 e a publicou em uma página na Internet.
- A Sony processou todos os hackers envolvidos, mas não realizou auto-crítica a respeito de seu erro de implementação.
- Como corrigir completamente o problema (sob o ponto de vista da Sony)?

O Diabo Está nos Detalhes: uma Análise da Quebra da Assinatura Digital do PlayStation 3

Conferência CAOS 2018

Prof. Luis Menasché Schechter
luisms@dcc.ufrj.br

Universidade Federal do Rio de Janeiro

5 de Outubro de 2018