Publishing Upper Half of RSA Decryption Exponent

Reynaldo Cáceres Villena
Orientador: Routo Terada

IME-USP

Maio de 2011

Agenda

1. Introdução
2. Motivação
3. RSA eficiente
4. Esquema ajudado por um servidor
5. Conclusões
6. Referências

1 - Introdução

1. Introdução
2. Motivação
3. RSA eficiente
4. Esquema ajudado por um servidor
5. Conclusões
6. Referências

Introdução

O criptossistema RSA foi proposto publicamente em 1978 e nomeado assim pelos seus criadores Ron **R**ivest, Adi **S**hamir e Leonard **A**dleman. RSA é o criptossistema de chave pública mais utilizado até hoje.

Criptossistema 1 RSA

Definimos $N = pq$ onde p e q são primos. Por definição da função Totiente do Euler temos que $\phi(N) = (p-1)(q-1)$.

KEYGEN: Escolha um e co-primo a $\phi(N) = (p-1)(q-1)$ e procure um d tal que $ed = 1 \mod \phi(N)$

KEYDIST: Publicar a chave $\langle N, e \rangle$ e manter em segredo $\langle N, d \rangle$

ENCRYPT: Para uma mensagem $M \in \mathbb{Z}_N$, o criptograma é $C = M^e \mod N$

DECRYPT: Para um criptograma C, a mensagem é $M = C^d \mod N$

Fato 1

Para um e pequeno, a metade superior de d pode ser calculado facilmente.

Prova.- A equação RSA $ed = 1 \mod \phi(N)$ pode ser expressada como $ed = 1 + k(N - (p+q) + 1)$, onde $L_e \approx L_k$ e $L_d \approx L_N$ .Em casos onde e é pequeno também k é pequeno. Portanto pode-se usar uma busca de força bruta para estimar o valor de d.

$$d = \frac{1+k(N+1)}{e} - \frac{k(p+q)}{e} \text{ , seja } d = d_0 + d_1 \text{ então temos:}$$

$$d_0 = \frac{1+k(N+1)}{e}, \text{ e } d_1 \equiv |d - d_0| < 2^{L_N/2}$$

Isto implica que o d_0 estimado é a metade superior de d e que $d_1 \equiv d \mod 2^{L_N/2}$

2 - Motivação

1 Introdução

2 Motivação

3 RSA eficiente

4 Esquema ajudado por um servidor

5 Conclusões

6 Referências

Motivação

Como foi explicado no fato 1 é possível determinar d_0 facilmente, esta estimação é muito estudada com o propósito de fazer ataques de chaves parciais expostas. A ideia é como aproveitar de maneira construtiva o fato 1.

- Podemos escolher os bits mais significativos de d para conseguir uma decifração mais eficiente?
- é possível escolher os bits menos significativos de d?

3 - RSA eficiente

1. Introdução
2. Motivação
3. RSA eficiente
4. Esquema ajudado por um servidor
5. Conclusões
6. Referências

RSA eficiente

Neste esquema o usuário pode escolher os bits mais significativos de d.
Como e é pequeno podemos dizer que $L_d = L_N$,
Seja $d = d_0 + d_1$ onde:

- d_0 é um inteiro de tamanho L_N onde os $(L_N/2 - L_e)$ bits mais significativos de d_0 e d são iguais e o resto são zeros.
- d_1 é um inteiro de tamanho $(L_N/2 + L_e)$ onde a maior parte de bits menos significativos de d é d_1.

Criptosistema 2 RSA

Escolhemos o e (expoente de encriptação), L_p, L_p (total de bits dos primos p e q) e d_0 (metade superior de bits para o expoente de decifração d).
KEYGEN: $(p, q, N, d) \leftarrow$ KEYGENALGOMSB (e, L_p, L_q, d_o).
KEYDIST: Publicar a chave $\langle N, e \rangle$ e manter em segredo $\langle N, d \rangle$
ENCRYPT: Para uma mensagem $M \in \mathbb{Z}_N$, o criptograma é $C = M^e mod N$
DECRYPT: Para um criptograma C, a mensagem é $M = C^d mod N$.

Algoritmo KEYGENALGOMSB

Entrada: (e, L_p, L_q, d_o), saída: parâmetros RSA(p, q, N, d)

1. Gerar um número primo aleatório p de tamanho L_p, tal que MDC$(p-1, e)$=1;
2. Gerar um número aleatório d_{pad} de tamanho menor o igual a $\frac{1}{2}L_{d_0}$;
3. $\tilde{d}_0 \leftarrow d_0 + d_{pad}$;
4. Gerar um número aleatório k de tamanho L_e, tal que MDC(k, e)=1;
5. $x \leftarrow e - [(k(p-1))^{-1} \mod e]$;
6. $y \leftarrow \lceil \frac{1}{k(p-1)}(e\tilde{d}_0 - 1) \rceil$;
7. $z \leftarrow \lceil \frac{1}{e}(y - x) \rceil$;
8. $w \leftarrow x + ze$;
9. Se $w + 1$ é primo e $L_w = L_q$ então continuar senão voltar à linha 2;
10. $q \leftarrow w + 1$;
11. $\tilde{d}_1 \leftarrow -\frac{1}{e}((e\tilde{d}_0 - 1) - k(p-1)w)$;
12. $d = \tilde{d}_0 + \tilde{d}_1$;
13. DEVOLVER (p, q, pq, d);

Correção do criptossistema 2 RSA

Teorema 1

A saída d gerada pelo algoritmo KEYGENALGOMSB(e, L_p, L_q, d_0) compartilha os bits $(\frac{1}{2}L_N\text{-}\ L_e)$ superiores da entrada d_0

A correção deste esquema depende sobre a correção do algoritmo KEYGENALGOMSB. Note que:

$$\begin{aligned} e\tilde{d}_0 - 1 - k(p-1)w &\equiv e\tilde{d}_0 - 1 - k(p-1)(x+ze) \\ &\equiv -1 + k(p-1)([k(p-1)]^{-1}) \mod e \\ &\equiv -1+1 \equiv 0 \mod e \end{aligned}$$

Portanto $e\tilde{d}_0 - 1 - k(p-1)w$ é multiplo de e

$$e\tilde{d}_0 - 1 - k(p-1)w = -e\tilde{d}_1$$

$$e(\tilde{d}_0 + \tilde{d}_1) = 1 + k(p-1)(q-1)$$

$$ed = 1 + k(p-1)(q-1)$$

Isto implica que o algoritmo gera os parâmetros do RSA.

Eficiência do criptossistema 3 RSA

A principal vantagem da cripstossistema 2 RSA está relacionado com a decriptação do RSA já que nós podemos escolher o $d_0 = 2^{L_N - 1}$.
Pelo algoritmo de Exponenciação sabemos que:

- $nro^{\overline{b1}} = (nro^{\overline{b}})^{10}.nro$
  - um bit 1 representa um quadrado e uma multiplicação $= u + 1$
- $nro^{\overline{b0}} = (nro^{\overline{b}})^{10}$
  - um bit 0 representa só um quadrado $= u$
- onde o u é o custo da operação 'potência de 2' com relação à multiplicação

Eficiência do cripstossistema 2 RSA

Para qualquer d vamos supor que a metade dos seus bits são 1's, portanto vamos ter $u.L_N + \frac{1}{2}L_N$ operações
para nosso criptossistema 2 vamos ter que $d_0 = 2^{L_N - 1}$ e supor a metade dos bits de d_1 são 1's. pelo qual temos:
$(u+1) + (\frac{1}{2}L_N - L_e - 1)u + (L_N + L_e)u + \frac{1}{2}(\frac{1}{2}L_N + L_e) = (u + \frac{1}{4})L_N + \frac{L_e}{2}$
A vantagem com respeito ao número menor de operações e:

$$1 - \frac{(u+\frac{1}{4})L_N + \frac{L_e}{2}}{u.L_N + \frac{1}{2}L_N} = \frac{1 - \frac{2L_e}{L_N}}{2(2u+1)}$$

sejá $e = 2^{16} + 1, L_p = L_q$, e $u = 1$

$L_p = L_q$	L_N	d_0	%
32	64	2^{63}	7.81
64	128	2^{127}	12.24
128	256	2^{255}	14.45
256	512	2^{511}	15.56
512	1024	2^{1023}	16.11

4 - Esquema ajudado por um servidor

1 Introdução

2 Motivação

3 RSA eficiente

4 Esquema ajudado por um servidor

5 Conclusões

6 Referências

Esquema ajudado por um servidor

Neste esquema vai-se usar o RSA regular.
Vamos considerar a seguinte situação onde Alice está usando um dispositivo de baixo poder de computação.

Criptossistema 3 RSA

Alice escolhe um e pequeno e fixa o tamanho dos primos p e q. Também deve escolher o d_1. Agora, os participantes são Alice, Paulo (servidor) e Beto(remetente).
KEYGEN: Alice gera os parâmetros de RSA usando o algoritmo $(p, q, N, d) \leftarrow \text{KEYGENALGOMSB}\ (e, L_p, L_q, d_o)$.
KEYDIST: Define $d_0 = d - d_1$. Alice publica a chave $\langle N, e \rangle$, da a Paulo a informação $\langle N, d_0 \rangle$ e mantem em segredo $\langle N, d_1 \rangle$
ENCRYPT: Bob encripta uma mensagem $M \in \mathbb{Z}_N$ como $C = M^e \mod N$
SERVER: Paulo calcula $V = C^{d_1} \mod N$ e envia a Alice $\langle V, C \rangle$
DECRYPT: Alice recebe $\langle V, C \rangle$ e calcula $M = VC_1 \mod N$.

Algoritmo KEYGENALGOLSB

Entrada: $(e, L_p, L_q, d_1 =)$, saída: parâmetros RSA(p, q, N, d)

1. Gerar um número primo aleatório p de tamanho $L_p = \frac{1}{2}L_N$, tal que MDC$(p-1, e)$=1 e $\frac{1}{2}(p-1) \bmod 2 = 1$;
2. Gerar um número inteiro aleatório k de tamanho $L_k = L_e$, tal que MCD$(k, e, 2) = 1$;
3. $x \leftarrow [\frac{1}{2}k(p-1)]^{-1} \mod (e.2^{L_q - L_e + 1})$;
4. $y \leftarrow [e(d_1 - 1)x] \mod (e.2^{L_q - L_e + 1})$;
5. $z \leftarrow \frac{1}{2}y + 1$;
6. Se z é primo e $L_z = L_q$ então continuar senão voltar à linha 2;
7. $q \leftarrow z$;
8. $w \leftarrow \frac{1}{e.2^{L_q - L_e + 1}}(-ed_1 + 1 + k(p-1)(q-1))$;
9. $d \leftarrow w.2^{L_q - L_e + 1} + d_1$;
10. DEVOLVER (p, q, pq, d);

Correção do criptossistema 3 RSA

Teorema

A saída d gerada pelo algoritmo KEYGENALGOLSB(e, L_p, L_q, d_1) compartilha os bits $(\frac{1}{2}L_N\text{-}\ L_e)$ inferiores da entrada d_1

A correção do esquema proposto depende da correção do algoritmo KEYGENALGOLSB. Podemos notar que:

$$\begin{aligned} ed &= ew2^{L_q - L_e + 1} + ed_1 \\ &= (-ed_1 + 1 + k(p-1)(q-1)) + ed_1 \\ &= 1 + k(p-1)(q-1) \end{aligned}$$

O qual representa a equação RSA $ed = 1 \mod N$. Isto prova a correção do criptossistema 3 RSA

Eficiência do criptossistema 3 RSA

A fase de encriptação é a mesma com respeito ao RSA regular portanto a eficiência é idêntico ao esquema RSA regular usando um 'e' pequeno. A vantagem é visto na fase de decifração. Como já vimos antes o costo da decifração é igual a $(u + \frac{1}{2})L_N)$.
Neste esquema estamos reduzindo a carga do dispositivo de Alice, mas não do servidor.
O dispositivo antes efetuava $(u + \frac{1}{2})L_N$ operações, mas agora só está efetuando $u(\frac{1}{2}L_N - L_e) + w_1$ onde w_1 é a quantidade de 1's em d_1

O dispositivo antes efetuava $(u + \frac{1}{2})L_N$ operações , mas agora só está efetuando $u(\frac{1}{2}L_N - L_e) + w_1$ Portanto Alice obtem uma vantagem em proporção ao número menor de operações

$$1 - \frac{u(\frac{1}{2}L_N - L_e) + w_1}{(u+\frac{1}{2})L_N} = \frac{u+1+\frac{2uL_e}{L_N} - \frac{2w_1}{L_N}}{2u+1}$$

Seja um cenário prático com $L_N = 1024, e = 2_{16} + 1, w_1 = 40$ e $u = 1$ a vantagem é:

$$\frac{u+1+\frac{2uL_e}{L_N} - \frac{2w_1}{L_N}}{2u+1} = 65.17\%$$

5 - Conclusões

1 Introdução

2 Motivação

3 RSA eficiente

4 Esquema ajudado por um servidor

5 Conclusões

6 Referências

Conclusões

- RSA eficiente: Escolher a metade superior de d pode ajudar a obter algumas vantagens na decifração.
- Esquema ajudado por um servidor: Escolher a metade inferior de d e dar a metade superior a um servidor, pode ajudar a diminuir o custo durante a decifração do RSA.

6 - Referências

1. Introdução
2. Motivação
3. RSA eficiente
4. Esquema ajudado por um servidor
5. Conclusões
6. Referências

Referéncias

- Subhamoy Maitra, Santanu Sarkar, and Sourav Sen Gupta *Publishing Upper Half of RSA Decryption Exponent*
- Boneh, D., Durfee, G., Frankel, Y.: - *Exposing an RSA Private Key given a Small Fraction of its Bits*
- Inês Barbedo - *O Sistema Criptofráfico RSA: Ataques e Variantes*
- Routo Terada - *Segurança de Dados: Criptografia en rede de computador 2ª Edição*
- Douglas R. Stinson - *Cryptography, Theory and Practice 3ª Edition*

dúvidas ???