Protokoły komunikacyjne

http://www.mimuw.edu.pl/~ sl/teaching/SEMPROT/

Podpisy cyfrowe

Artur Cichocki

Wydział Matematyki, Informatyki i Mechaniki
Uniwersytet Warszawski
email: A.Cichocki@zodiac.mimuw.edu.pl
źródła TEX: http://rainbow.mimuw.edu.pl/~ ac181253/

Wstęp

- Paradygmat podpisów RSA.
- Schemat standardowy trudno udowodnić.
- Inne schematy zgodne z paradygmatem i udowodnialne (FDH, PSS, PSS-R).




Schemat podpisu cyfrowego

$DS = (\mathcal{K}, \mathcal{S}, \mathcal{V})$ składa się z trzech algorytmów:

generatora kluczy $\mathcal{K}$ algorytm randomizowany, który zwraca parę (pk, sk) kluczy, publiczny klucz i odpowiadający mu klucz prywatny (piszemy $(pk, sk) \xleftarrow{R} \mathcal{K}$).

Schemat podpisu cyfrowego

$DS = (\mathcal{K}, \mathcal{S}, \mathcal{V})$ składa się z trzech algorytmów:

generatora kluczy $\mathcal{K}$ algorytm randomizowany, który zwraca parę (pk, sk) kluczy, publiczny klucz i odpowiadający mu klucz prywatny (piszemy $(pk, sk) \stackrel{R}{\leftarrow} \mathcal{K}$).

algorytm podpisujący $\mathcal{S}$ (być może randomizowany) biorący sk i wiadomość M, a zwracający δ (piszemy $\delta \leftarrow \mathcal{S}_{sk}(M)$).

Schemat podpisu cyfrowego

$DS = (\mathcal{K}, \mathcal{S}, \mathcal{V})$ składa się z trzech algorytmów:

generatora kluczy $\mathcal{K}$ algorytm randomizowany, który zwraca parę (pk, sk) kluczy, publiczny klucz i odpowiadający mu klucz prywatny (piszemy $(pk, sk) \stackrel{R}{\leftarrow} \mathcal{K}$).

algorytm podpisujący $\mathcal{S}$ (być może randomizowany) biorący sk i wiadomość M, a zwracający δ (piszemy $\delta \leftarrow \mathcal{S}_{sk}(M)$).

algorytm weryfikujący $\mathcal{V}$ deterministyczny, przyjmujący pk, M' i δ' dla M, a zwracający jeden bit $(d \leftarrow \mathcal{V}_{pk}(M', \delta'))$.

Schemat podpisu cyfrowego

$DS = (\mathcal{K}, \mathcal{S}, \mathcal{V})$ składa się z trzech algorytmów:

generatora kluczy $\mathcal{K}$ algorytm randomizowany, który zwraca parę (pk, sk) kluczy, publiczny klucz i odpowiadający mu klucz prywatny (piszemy $(pk, sk) \stackrel{R}{\leftarrow} \mathcal{K}$).

algorytm podpisujący $\mathcal{S}$ (być może randomizowany) biorący sk i wiadomość M, a zwracający δ (piszemy $\delta \leftarrow \mathcal{S}_{sk}(M)$).

algorytm weryfikujący $\mathcal{V}$ deterministyczny, przyjmujący pk, M' i δ' dla M, a zwracający jeden bit $(d \leftarrow \mathcal{V}_{pk}(M', \delta'))$.

Rozprowadzanie kluczy i dostarczanie wiadomości (M, δ).

Wstecz
Zamknij

Rodzaje ataków

tylko klucz Przeciwnik zna tylko klucz publiczny i może sprawdzić poprawność podpisu.




Rodzaje ataków

tylko klucz Przeciwnik zna tylko klucz publiczny i może sprawdzić poprawność podpisu.

znane podpisy Przeciwnik zna klucz publiczny i ma pary (wiadomość, podpis), podpis legalnie wyprodukowany.

Rodzaje ataków

tylko klucz Przeciwnik zna tylko klucz publiczny i może sprawdzić poprawność podpisu.

znane podpisy Przeciwnik zna klucz publiczny i ma pary (wiadomość, podpis), podpis legalnie wyprodukowany.

wybór wiadomości Przeciwnik może pytać wyrocznię o podpis wybranych przez siebie wiadomości.

Poziomy sukcesów

oszustwo egzystencjalne Przeciwnik może podrobić podpis jednej wiadomości, ale nie koniecznie może wybierać wiadomość.

Poziomy sukcesów

oszustwo egzystencjalne Przeciwnik może podrobić podpis jednej wiadomości, ale nie koniecznie może wybierać wiadomość.

selektywne Dodatkowo może wybrać jakąś wiadomość.

Poziomy sukcesów

oszustwo egzystencjalne Przeciwnik może podrobić podpis jednej wiadomości, ale nie koniecznie może wybierać wiadomość.

selektywne Dodatkowo może wybrać jakąś wiadomość.

uniwersalne Dowolną wiadomość, ale nie pozna klucza prywatnego.

Poziomy sukcesów

oszustwo egzystencjalne Przeciwnik może podrobić podpis jednej wiadomości, ale nie koniecznie może wybierać wiadomość.

selektywne Dodatkowo może wybrać jakąś wiadomość.

uniwersalne Dowolną wiadomość, ale nie pozna klucza prywatnego.

totalne Może policzyć klucz prywatny.

Pojęcie bezpieczeństwa

- Eksperyment $Exp_{DS,F}^{uf-cma}$

  Niech $(pk, sk) \xleftarrow{R} \mathcal{K}$
  Niech $(M, \sigma) \leftarrow F^{\mathcal{S}_{sk}(\cdot)}(pk)$
  Jeżeli $\mathcal{V}_{pk}(M, \delta) = 1$ i wyrocznia nie była pytana o M wtedy zwróć 1, w przeciwnym przypadku 0.




Pojęcie bezpieczeństwa

- Eksperyment $Exp_{DS,F}^{uf-cma}$

  Niech $(pk, sk) \xleftarrow{R} \mathcal{K}$
  Niech $(M, \sigma) \leftarrow F^{\mathcal{S}_{sk}(\cdot)}(pk)$
  Jeżeli $\mathcal{V}_{pk}(M, \delta) = 1$ i wyrocznia nie była pytana o M wtedy zwróć 1, w przeciwnym przypadku 0.

- Niech $DS = (\mathcal{K}, \mathcal{S}, \mathcal{V})$, a F oznacza fałszerza. $Adv_{DS,F}^{uf-cma}$ prawdopodobieństwo, że $Exp_{DS,F}^{uf-cma}$ zwróci 1, wtedy

Pojęcie bezpieczeństwa

- Eksperyment $Exp_{DS,F}^{uf-cma}$

  Niech $(pk, sk) \xleftarrow{R} \mathcal{K}$
  Niech $(M, \sigma) \leftarrow F^{\mathcal{S}_{sk}(\cdot)}(pk)$
  Jeżeli $\mathcal{V}_{pk}(M, \delta) = 1$ i wyrocznia nie była pytana o M wtedy zwróć 1, w przeciwnym przypadku 0.

- Niech $DS = (\mathcal{K}, \mathcal{S}, \mathcal{V})$, a F oznacza fałszerza. $Adv_{DS,F}^{uf-cma}$ prawdopodobieństwo, że $Exp_{DS,F}^{uf-cma}$ zwróci 1, wtedy

$$Adv_{DS}^{uf-cma}(t, q, \mu) = \max_{F}\{Adv_{DS,F}^{uf-cma}\}.$$

Generowanie klucza

- Algorytm $\mathcal{K}$

  Wybierz losowo dwie różne liczy pierwsze p i q (po $k/2$ bitów)

  $N \leftarrow pg$; $e \xleftarrow{R} Z^*_{\varphi(N)}$; $d \leftarrow e^{-1} \bmod \varphi(N)$

  $pk \leftarrow (N, e)$; $sk \leftarrow (N, d)$

  Zwróć pk, sk.

Generowanie klucza

- Algorytm $\mathcal{K}$

  Wybierz losowo dwie różne liczy pierwsze p i q (po $k/2$ bitów)

  $N \leftarrow pg$; $e \stackrel{R}{\leftarrow} Z^*_{\varphi(N)}$; $d \leftarrow e^{-1} \bmod \varphi(N)$

  $pk \leftarrow (N, e)$; $sk \leftarrow (N, d)$

  Zwróć pk, sk.

- Permutacje $x \in Z^*_N$
  - $RSA_{N,e}(x) = x^e \bmod N = y$

Wstecz

Zamknij

Generowanie klucza

- Algorytm $\mathcal{K}$

  Wybierz losowo dwie różne liczy pierwsze p i q (po $k/2$ bitów)

  $N \leftarrow pg$; $e \stackrel{R}{\leftarrow} Z^*_{\varphi(N)}$; $d \leftarrow e^{-1} \bmod \varphi(N)$

  $pk \leftarrow (N, e)$; $sk \leftarrow (N, d)$

  Zwróć pk, sk.

- Permutacje $x \in Z^*_N$
  - $RSA_{N,e}(x) = x^e \bmod N = y$
  - $RSA_{N,d}(y) = y^d \bmod N = RSA^{-1}_{N,e}(y) = x$

Generowanie klucza

- Algorytm $\mathcal{K}$

  Wybierz losowo dwie różne liczy pierwsze p i q (po $k/2$ bitów)

  $N \leftarrow pg$; $e \xleftarrow{R} Z^*_{\varphi(N)}$; $d \leftarrow e^{-1} \bmod \varphi(N)$

  $pk \leftarrow (N, e)$; $sk \leftarrow (N, d)$

  Zwróć pk, sk.

- Permutacje $x \in Z^*_N$
  - $RSA_{N,e}(x) = x^e \bmod N = y$
  - $RSA_{N,d}(y) = y^d \bmod N = RSA^{-1}_{N,e}(y) = x$

- Zakładamy, że RSA jest permutacją jednokierunkową.

Podpisy z zapadką

- Algorytm

Algorytm $\mathcal{S}_{N,d}(M)$	Algorytm $\mathcal{V}_{N,e}(M)$
$x \leftarrow M^d \bmod N$	$M' \leftarrow x^e \bmod N$
Zwróć x	Jeżeli $M = M'$ zwróć 1, wpp 0.

Podpisy z zapadką

- Algorytm

Algorytm $\mathcal{S}_{N,d}(M)$	Algorytm $\mathcal{V}_{N,e}(M)$
$x \leftarrow M^d \bmod N$	$M' \leftarrow x^e \bmod N$
Zwróć x	Jeżeli $M = M'$ zwróć 1, wpp 0.

- Podpis, a potem wiadomość

Fałszerz $F^{\mathcal{S}_{N,e}(\cdot)}(N, e)$

$x \stackrel{R}{\leftarrow} Z_N^*$; $M \leftarrow x^e \bmod N$

Zwróć (M, x)

Podpisy z zapadką

- Algorytm

Algorytm $\mathcal{S}_{N,d}(M)$	Algorytm $\mathcal{V}_{N,e}(M)$
$x \leftarrow M^d \bmod N$	$M' \leftarrow x^e \bmod N$
Zwróć x	Jeżeli $M = M'$ zwróć 1, wpp 0.

- Podpis, a potem wiadomość

  Fałszerz $F^{\mathcal{S}_{N,e}(\cdot)}(N, e)$

  $x \stackrel{R}{\leftarrow} Z_N^*$; $M \leftarrow x^e \bmod N$

  Zwróć (M, x)

- Multiplikatywność RSA

$$x^e \equiv (x_1 x_2)^e \equiv x_1^e x_2^e \equiv M_1 M_2 \equiv M \pmod{N}$$

Fałszerz $F^{\mathcal{S}_{N,e}(\cdot)}(N, e)$

$M_1 \stackrel{R}{\leftarrow} Z_N^* - \{1, M\}$; $M_2 \leftarrow M M_1^{-1} \bmod N$

$x_1 \leftarrow \mathcal{S}_{N,e}(M_1)$; $x_2 \leftarrow \mathcal{S}_{N,e}(M_2)$

$x \leftarrow x_1 x_2 \bmod N$

Zwróć (M, x)

Mieszaj i odwróć

- Algorytm

Algorytm $\mathcal{S}_{N,d}(M)$	Algorytm $\mathcal{V}_{N,e}(M)$
$y \leftarrow Hash(M)$	$y \leftarrow Hash(M)$
$x \leftarrow y^d \bmod N$	$y' \leftarrow x^e \bmod N$
Zwróć x	Jeżeli $y = y'$ zwróć 1, wpp 0.

Mieszaj i odwróć

- Algorytm

Algorytm $\mathcal{S}_{N,d}(M)$	Algorytm $\mathcal{V}_{N,e}(M)$
$y \leftarrow Hash(M)$	$y \leftarrow Hash(M)$
$x \leftarrow y^d \bmod N$	$y' \leftarrow x^e \bmod N$
Zwróć x	Jeżeli $y = y'$ zwróć 1, wpp 0.

- Odporność na poprzednie ataki.
  - $x^e \bmod N = Hash(M)$
  - $Hash(M_1)Hash(M_2) \equiv Hash(M) \pmod N$

Mieszaj i odwróć

- Algorytm

Algorytm $\mathcal{S}_{N,d}(M)$	Algorytm $\mathcal{V}_{N,e}(M)$
$y \leftarrow Hash(M)$	$y \leftarrow Hash(M)$
$x \leftarrow y^d \bmod N$	$y' \leftarrow x^e \bmod N$
Zwróć x	Jeżeli $y = y'$ zwróć 1, wpp 0.

- Odporność na poprzednie ataki.
  - $x^e \bmod N = Hash(M)$
  - $Hash(M_1)Hash(M_2) \equiv Hash(M) \pmod N$

- Kolizje $Hash(M_1) = Hash(M_2)$

Mieszaj i odwróć

- Algorytm

Algorytm $\mathcal{S}_{N,d}(M)$	Algorytm $\mathcal{V}_{N,e}(M)$
$y \leftarrow Hash(M)$	$y \leftarrow Hash(M)$
$x \leftarrow y^d \bmod N$	$y' \leftarrow x^e \bmod N$
Zwróć x	Jeżeli $y = y'$ zwróć 1, wpp 0.

- Odporność na poprzednie ataki.
  - $x^e \bmod N = Hash(M)$
  - $Hash(M_1)Hash(M_2) \equiv Hash(M) \pmod N$

- Kolizje $Hash(M_1) = Hash(M_2)$

  Fałszerz $F^{\mathcal{S}_{N,e}(\cdot)}(N, e)$
  $x \leftarrow \mathcal{S}_{N,e}(M_1)$
  Zwróć (M_2, x)

PKCS #1

- Implementacja funkcji mieszającej.

$$\frac{N-\varphi(N)}{N} = 1 - \frac{(p-1)(q-1)}{pg} = \frac{p+g-1}{pg} < \frac{2^{1+k/2}}{2^{k-1}} = 4 \cdot 2^{-k/2}$$

PKCS #1

- Implementacja funkcji mieszającej.

$$\frac{N-\varphi(N)}{N} = 1 - \frac{(p-1)(q-1)}{pg} = \frac{p+g-1}{pg} < \frac{2^{1+k/2}}{2^{k-1}} = 4 \cdot 2^{-k/2}$$

- PCKS-Hash(M) = 0x 00 01 FF FF $\cdots$ FF FF 00 || h(M)

PKCS #1

- Implementacja funkcji mieszającej.

$$\frac{N-\varphi(N)}{N} = 1 - \frac{(p-1)(q-1)}{pg} = \frac{p+g-1}{pg} < \frac{2^{1+k/2}}{2^{k-1}} = 4 \cdot 2^{-k/2}$$

- PCKS-Hash(M) = 0x 00 01 FF FF $\cdots$ FF FF 00 || h(M)
- Algorytm $\mathcal{S}_{N,d}(M)$
  $y \leftarrow PCKS\text{-}Hash(M)$
  $x \leftarrow y^d \bmod N$
  Zwróć x

  Algorytm $\mathcal{V}_{N,e}(M)$
  $y \leftarrow PCKS\text{-}Hash(M)$
  $y' \leftarrow x^e \bmod N$
  Jeżeli $y = y'$ zwróć 1, wpp 0.

PKCS #1

- Implementacja funkcji mieszającej.

$$\frac{N-\varphi(N)}{N} = 1 - \frac{(p-1)(q-1)}{pg} = \frac{p+g-1}{pg} < \frac{2^{1+k/2}}{2^{k-1}} = 4 \cdot 2^{-k/2}$$

- PCKS-Hash(M) = 0x 00 01 FF FF $\cdots$ FF FF 00 || h(M)
- Algorytm $\mathcal{S}_{N,d}(M)$ | Algorytm $\mathcal{V}_{N,e}(M)$

  $y \leftarrow PCKS\text{-}Hash(M)$ | $y \leftarrow PCKS\text{-}Hash(M)$

  $x \leftarrow y^d \bmod N$ | $y' \leftarrow x^e \bmod N$

  Zwróć x | Jeżeli $y = y'$ zwróć 1, wpp 0.
- Dla SHA-1 $S = \{\text{PCKS-Hash(M)} : M \in \{0,1\}^*\}$, $|S| \leq 2^{160}$

$$\frac{|S|}{|Z_N^*|} \leq \frac{2^{160}}{2^{1023}} = \frac{1}{2^{863}}$$

FDH

- Mając "idealną" funkcje mieszającą.

FDH

- Mając "idealną" funkcje mieszającą.
- Funkcja mieszająca jako wyrocznia.

FDH

- Mając "idealną" funkcje mieszającą.
- Funkcja mieszająca jako wyrocznia.
- Algorytm (poprzedni 10).

Algorytm $\mathcal{S}^H_{N,d}(M)$	Algorytm $\mathcal{V}^H_{N,e}(M)$
$y \leftarrow H(M)$	$y \leftarrow H(M)$
$x \leftarrow y^d \bmod N$	$y' \leftarrow x^e \bmod N$
Zwróć x	Jeżeli $y = y'$ zwróć 1, wpp 0.

- Nowy model bezpieczeństwa (poprzedni 6).

  Eksperyment $Exp^{ro}(DS, F)$

  Niech $((N, e), (N, d)) \stackrel{R}{\leftarrow} \mathcal{K}$
  Wybrać losowo $H : \{0,1\}^* \rightarrow Z_N^*$
  Niech $(M, x) \leftarrow F^{H, \mathcal{S}^H_{N,d}(\cdot)}(N, e)$
  Jeżeli $\mathcal{V}^H_{(N,e)}(M, \delta) = 1$ i wyrocznia nie była pytana o M wtedy zwróć 1, w przeciwnym przypadku 0.




Wstecz
Zamknij

- Nowy model bezpieczeństwa (poprzedni 6).

  Eksperyment $Exp^{ro}(DS, F)$

  Niech $((N, e), (N, d)) \stackrel{R}{\leftarrow} \mathcal{K}$
  Wybrać losowo $H : \{0,1\}^* \rightarrow Z_N^*$
  Niech $(M, x) \leftarrow F^{H, \mathcal{S}^H_{N,d}(\cdot)}(N, e)$
  Jeżeli $\mathcal{V}^H_{(N,e)}(M, \delta) = 1$ i wyrocznia nie była pytana o M wtedy zwróć 1, w przeciwnym przypadku 0.

- Szanse fałszerstwa.

$$Adv^{uf-cma}_{DS}(t, q_{sig}, q_{hash}, \mu) = \max_F \{Adv^{uf-cma}_{DS,F}\}$$

- Nowy model bezpieczeństwa (poprzedni 6).

  Eksperyment $Exp^{ro}(DS, F)$

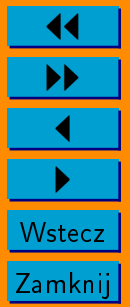
  Niech $((N, e), (N, d)) \stackrel{R}{\leftarrow} \mathcal{K}$
  Wybrać losowo $H : \{0,1\}^* \rightarrow Z_N^*$
  Niech $(M, x) \leftarrow F^{H, \mathcal{S}^H_{N,d}(\cdot)}(N, e)$
  Jeżeli $\mathcal{V}^H_{(N,e)}(M, \delta) = 1$ i wyrocznia nie była pytana o M wtedy zwróć 1, w przeciwnym przypadku 0.

- Szanse fałszerstwa.

$$Adv^{uf-cma}_{DS}(t, q_{sig}, q_{hash}, \mu) = \max_F \{Adv^{uf-cma}_{DS,F}\}$$

- Niech DS będzie schematem FDH-RSA z parametrem k. Wtedy dla $q_{hash} \geq 1 + q_{sig}$ i $t' = t + q_{hash} \cdot O(k^3)$ zachodzi

$$Adv^{uf-cma}_{DS}(t, q_{sig}, q_{hash}, \mu) \leq q_{hash} \cdot Adv^{owf}_{RSA}(t').$$



PSS0

- Algorytm

Algorytm $\mathcal{S}^H_{N,d}(M)$	Algorytm $\mathcal{V}^H_{N,e}(M)$
$r \xleftarrow{R} \{0,1\}^s$	Sparsuj δ jako (r, x) gdzie $\|r\| = s$
$y \leftarrow H(r\|\|M)$	$y \leftarrow H(r\|\|M)$
$x \leftarrow y^d \bmod N$	$y' \leftarrow x^e \bmod N$
Zwróć (r, x)	Jeżeli $y = y'$ zwróć 1, wpp 0.

PSS0

- Algorytm

Algorytm $\mathcal{S}^H_{N,d}(M)$	Algorytm $\mathcal{V}^H_{N,e}(M)$
$r \xleftarrow{R} \{0,1\}^s$	Sparsuj δ jako (r,x) gdzie $\|r\| = s$
$y \leftarrow H(r\|\|M)$	$y \leftarrow H(r\|\|M)$
$x \leftarrow y^d \bmod N$	$y' \leftarrow x^e \bmod N$
Zwróć (r,x)	Jeżeli $y = y'$ zwróć 1, wpp 0.

- Niech DS będzie schematem PSS0 z parametrem k i s. Wtedy dla $q_{hash} \geq 1 + q_{sig}$ i $t' = t + q_{hash} \cdot O(k^3)$ zachodzi

$$Adv^{uf-cma}_{DS}(t, q_{sig}, q_{hash}, \mu) \leq Adv^{owf}_{RSA}(t') + \frac{(q_{hash} - 1) \cdot q_{sig}}{2^s}.$$




PSS

- Algorytm

Algorytm $\mathcal{S}^{g,h}_{N,d}(M)$	Algorytm $\mathcal{V}^{g,h}_{N,e}(M)$
$r \stackrel{R}{\leftarrow} \{0,1\}^{k_0}$; $w \leftarrow H(M\|\|r)$	$y \leftarrow x^e \bmod N$
$r^* \leftarrow g_1(w) \oplus r$	Sparsuj y jako $b\|\|w\|\|r^*\|\|\gamma$ gdzie
$y \leftarrow 0\|\|w\|\|r^*\|\|g_2(w)$	$\|b\| = 1, \|w\| = k_1, \|r^*\| = k_0$
$x \leftarrow y^d \bmod N$	$r \leftarrow r^* \oplus g_1(w)$
Zwróć x	Jeżeli $h(M\|\|r) = w$ i $g_2(w) = \gamma$
	i $b = 0$ wtedy zwróć 1, wpp 0.

- Jeszcze lepiej

$$Adv^{uf-cma}_{DS}(t, q_{sig}, q_{hash}, \mu) \leq Adv^{owf}_{RSA}(t') + 3(q_{hash} - 1)^2 \cdot (2^{-k_0} + 2^{-k_1}).$$

PSS

- Algorytm

Algorytm $\mathcal{S}^{g,h}_{N,d}(M)$	Algorytm $\mathcal{V}^{g,h}_{N,e}(M)$
$r \xleftarrow{R} \{0,1\}^{k_0}$; $w \leftarrow H(M\|\|r)$	$y \leftarrow x^e \bmod N$
$r^* \leftarrow g_1(w) \oplus r$	Sparsuj y jako $b\|\|w\|\|r^*\|\|\gamma$ gdzie
$y \leftarrow 0\|\|w\|\|r^*\|\|g_2(w)$	$\|b\| = 1, \|w\| = k_1, \|r^*\| = k_0$
$x \leftarrow y^d \bmod N$	$r \leftarrow r^* \oplus g_1(w)$
Zwróć x	Jeżeli $h(M\|\|r) = w$ i $g_2(w) = \gamma$
	i $b = 0$ wtedy zwróć 1, wpp 0.

- Jeszcze lepiej

$$Adv^{uf-cma}_{DS}(t, q_{sig}, q_{hash}, \mu) \leq Adv^{owf}_{RSA}(t') + 3(q_{hash} - 1)^2 \cdot (2^{-k_0} + 2^{-k_1}).$$

- PSS-R = odzyskiwanie wiadomości z podpisu.

El Gamal

klucz publiczny (y, p, g), gdzie $y = g^x \bmod p$, p pierwsza a g generator w Z_p^*.

klucz prywatny x takie, że $y = g^x \bmod p$.

podpis podpisem wiadomości m jest para (r, s) taka, że $r \neq 0$, $s \neq p - 1$ i $g^m = y^r r^s \bmod p$.

weryfikacja sprawdzenie czy $g^m = y^r r^s \bmod p$.

El Gamal

klucz publiczny (y, p, g), gdzie $y = g^x \bmod p$, p pierwsza a g generator w Z_p^*.

klucz prywatny x takie, że $y = g^x \bmod p$.

podpis podpisem wiadomości m jest para (r, s) taka, że $r \neq 0$, $s \neq p-1$ i $g^m = y^r r^s \bmod p$.

weryfikacja sprawdzenie czy $g^m = y^r r^s \bmod p$.

szczegóły podpisywania wybrać k losowo, $r = g^k \bmod p$, wtedy $g^m = y^r r^s = g^{xr+ks} \bmod p$, więc $s = (m - xr)k^{-1} \bmod p - 1$.

El Gamal

klucz publiczny (y, p, g), gdzie $y = g^x \bmod p$, p pierwsza a g generator w Z_p^*.

klucz prywatny x takie, że $y = g^x \bmod p$.

podpis podpisem wiadomości m jest para (r, s) taka, że $r \neq 0$, $s \neq p-1$ i $g^m = y^r r^s \bmod p$.

weryfikacja sprawdzenie czy $g^m = y^r r^s \bmod p$.

szczegóły podpisywania wybrać k losowo, $r = g^k \bmod p$, wtedy $g^m = y^r r^s = g^{xr+ks} \bmod p$, więc $s = (m - xr)k^{-1} \bmod p - 1$.

klapa Schemat pozwala na *oszustwo egzystencjalne* przy ataku *znane podpisy*.

Rabin

klucz publiczny $n = pq$.

klucz prywatny p i q.

podpisywanie $s = \sqrt{m} \bmod n$.

weryfikacja $s^2 = m \bmod n$.

Rabin

klucz publiczny $n = pq$.

klucz prywatny p i q.

podpisywanie $s = \sqrt{m} \bmod n$.

weryfikacja $s^2 = m \bmod n$.

Uups Schemat pozwala na *oszustwo egzystencjalne* przy ataku *tylko klucz*.

Rabin

klucz publiczny $n = pq$.

klucz prywatny p i q.

podpisywanie $s = \sqrt{m} \bmod n$.

weryfikacja $s^2 = m \bmod n$.

Uups Schemat pozwala na *oszustwo egzystencjalne* przy ataku *tylko klucz*.

Uuuuups Schemat pozwala na *oszustwo totalne* przy ataku *wybór wiadomości*.

Dziękuje za uwagę