Cvičení (co jsme se již naučili)

Na základě znalostí, které již máte si předvedeme, jak lze za jistých okolností získat podpis nějaké osoby pod (námi připravený) text, aniž by daná osoba vědomě tento konkrétní text podepsala.

Pro jednoduchost a srozumitelnost výkladu si vše předvedeme na následujícím jednoduchém podpisovém schématu.

I.1 Podpisové schéma MFF UK

V podstatě se jedná o klasické podpisové schéma, založená na RSA, kde je však vynechána hashovací funkce a text není formátován podle PKCS #1 (verze 1.5,2.0,2.1), ale podle námi zadaných pravidel, která označíme jako MFFUK #1.0.

I.1.1 – použitý asymetrický algoritmus

Vyjdeme z klasického RSA. Zvolíme prvočísla p a q a vypočteme

$N = p*q$

$\Phi(N) = (p-1)*(q-1)$

Dále zvolíme náhodné číslo e, kde

$1 < e < \Phi(N)$, takové, že e a $\Phi(N)$ jsou nesoudělná.

Vypočteme číslo d takové, že

$1 < d < \Phi(N)$ a

$e*d \equiv 1 \mod \Phi(N)$

Dvojici (N,d) nazveme soukromý klíč (resp. data na vytváření podpisu) a (N,e) veřejný klíč (resp. data na ověření podpisu).

I.1.2 Formátování zprávy (MFFUK#1.0)

Zprávu M překódujeme nejprve do číselného tvaru. K tomu použijeme některou vhodnou převodovou tabulku. Např. tuto tabulku:

	0	1	2	3	4	5	6	7	8	9
6	**0**	**Mezera**	**2**	**3**	**4**	**A**	**B**	**C**	**D**	**E**
7	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**	**O**
8	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**
9	**Z**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**

Zprávu M pak zformátujeme do posloupnosti čísel pevné délky (délka bude rovna délce modulu N). K tomu použijeme vlastní formátování, která pracovně nazveme MFFUK#1.0:

Formátování MFFUK#1.0 :

1) Má-li modul délku k, budeme zprávu v dekadickém tvaru dělit na skupiny délky k-1.
2) Všechny skupiny musí mít délku k-1, nemá-li poslední skupina tuto délku, doplníme ji zprava příslušným počtem nul.
3) Skupiny nyní doplníme zleva jednou nulou. Délka každé skupiny je tedy rovna k.
4) Výsledek po podpisové transformaci má délku rovnou maximálně k, nemá-li ji doplníme výsledek zleva nulami.

Získaný výsledek po formátování M označme $M= m_1\ m_2\ m_3\ \ldots..$

I.1.3 Podpis a ověření zprávy M

Podpisem zprávy M pak nazveme řetězec
$P = C_1\ C_2\ C_3$ ..... , kde
$C_1 \equiv m_1^d \bmod N$, $C_2 \equiv m_2^d \bmod N$, $C_3 \equiv m_3^d \bmod N$ ..... $C_i \equiv m_i^d \bmod N$

Ověření podpisu zprávy M se pak provede tak, že vypočteme pomocí dat na ověření podpisu následující výrazy
$V_1 \equiv C_1^e \bmod N$, $V_2 \equiv C_2^e \bmod N$, $V_3 \equiv C_3^e \bmod N$ .... $V_i \equiv C_i^e \bmod N$

Pokud $V_i = m_i$ pro všechna i, řekneme, že ověření podpisu bylo úspěšně provedeno.

Pokud podepisující osoba dokáže udržet svá data na podepisování v tajnosti (a čísla p a q byla dostatečně velká), pak je výpočetně složité ze znalosti podpisu zprávy a dat na ověření podpisu vypočítat soukromý – podepisovací klíč.

I.2 Zneužití

Ukážeme, jak získat podpis majitele soukromého klíče (N,d) pod zprávu M, aniž by to dotyčný majitel věděl.

Celá myšlenka je založena na tom, že RSA je multiplikativní vzhledem k násobení ($\forall a,b \in Z , k \in N : (ab)^k \equiv a^k b^k \bmod N$).

Mějme zprávu M, ke které chceme získat podpis nějaké osoby (Boba), tj. hodnotu M^d mod N. Bobovi předložíme místo vlastní hodnoty M, kterou by Bob mohl odmítnout podepsat, (zdánlivě) náhodnou hodnotu X. Tuto hodnotu X však předem pečlivě připravíme a to jako $M\ c^e \bmod N$. Zde c je náhodně zvolená veličina, (N,e) veřejný klíč Boba, M zpráva. Pokud Bob takovýto zdánlivě „nesmyslný“ text podepíše (např. v rámci autentizace) a my se k výsledku dostaneme (např. v rámci autentizace), pak jsme schopni poměrně jednoduše vypočítat podpis Boba pro zprávu M.

I.2.1 Příklad - část 1 - klíče

Vše si ukážeme na konkrétním příkladě:
Nejprve vytvoříme nějaký Bobův soukromý a veřejný klíč.

Zvolíme prvočísla: p=47, q=71,
Spočteme modul : N= p*q = 47*71 = 3337
a dále: $\Phi(N) = (p-1)*(q-1) = 46*70 = 3220$
Zvolíme veřejný exponent e (nesmí mít společné dělitele s 3220) , volíme např. 79
Spočteme soukromý exponent d:

d …… $79*d \equiv 1 \bmod 3220$

$d \equiv 79^{-1} \bmod 3220$

d = 1019 (k výpočtu použijeme Eukleidův algoritmus)

Získali jsme:

e …….. veřejný klíč (3337, 79) ,
d …..…... soukromý klíč (3337, 1019)

I.2.2 Příklad – část 2 - text

Předpokládejme, že chceme získat Bobův podpis zprávy M=DLUH JE 10 USD
Nejprve si převedeme text zprávy M pomocí kódové tabulky do číselné posloupnosti .
M= D L U H J E 1 0 U S D
M= 68 76 85 72 61 74 69 61 91 60 61 85 83 68

M dále zformátujeme podle pravidla MFFUK#1.0 na bloky m_1 m_2 m_3 …..
M = m_1 m_2 m_3 ….. = 0687 0685 0726 0174 0696 0191 0606 0185 0836 0800

Předpokládejme, že Bob má k dispozici program / prohlížeč, který by mu tuto zprávu zobrazil jako : DLUH JE 10 USD

Kdyby Bob podepsal pomocí svého soukromého klíče d tuto přímo tuto zprávu M
(tj. spočte $M^d \bmod N$, pro N=3337, d=1019) dostaneme (viz pomocný program RSAM):
1592 0585 1494 3172 644 3080 0647 1855 0707 1740 (*).

Naším cílem je tedy získat tuto posloupnost jiným způsobem, tedy bez toho, aby Bob podepsala přímo zformátovanou zprávu M.

K tomuto účelu si připravíme jinou – pomocnou zprávu.

I.2.3 Příklad – část 3 – pomocný text

Zvolíme nějaké libovolné číslo c, např. 105 a dále spočteme číslo $x \equiv c^e \bmod N$.
Pro konkrétní hodnoty Bobova veřejného klíče dostaneme $x \equiv 105^{79} \bmod 3337 \equiv 193$.
Dále připravíme k podpisu (zdánlivě) náhodnou „Bobovi nic neříkající“ hodnotu $M\,c^e \bmod N$.

Pro naše konkrétní hodnoty spočteme (např. využitím standardní kalkulačky ve Windows):
M = m_1 m_2 m_3 ….. = 687 685 726 174 696 191 606 185 836 800
$M\,c^e \bmod N = m_1\,c^e \bmod N \quad m_2\,c^e \bmod N \quad m_3\,c^e \bmod N$ ….. =
687*193 mod 3337 685*193 mod 3337 726*193 mod 3337 …..

M	0687	0685	0726	0174	696	0191	0606	0185	0836	800
$M\,c^e \bmod N$	2448	2062	3301	0212	848	0156	0163	2335	1172	898

Bob by při prohlížení tohoto textu svým prohlížečem viděl text, který zdánlivě nemá žádný smysl.

I.2.4 Příklad – část 4 – útok

Vraťme se k našemu příkladu. Pomocný text, který jsme si připravili, je tento:
$M\,c^e \bmod N$ 2448 2062 3301 0212 848 0156 0163 2335 1172 898
Takto připravený text předložíme Bobovi k podpisu.

a) Bud v rámci autentizace, kde místo náhodného řetězce pošleme tento text
b) Předložíme mu je k podpisu přímo (např. v rámci výuky – jak se elektronicky podepisovat…). Bob vidí nesmyslný obsah a text proto klidně podepíše. Bob tedy spočte $(M*c^e)^d \bmod N$ a dostane (viz program RSAM):

0310 1359 0031 2697 880 3048 1195 1229 0821 2502

Podívejme se, co po podpisu připraveného textu dostaneme (řada kroků je vynechána nebo jen naznačena)
$(M\ c^e \bmod N)^d \bmod N \equiv M^d * c^{ed} \bmod N \equiv M^d * c \bmod N$ (využito $e*d \equiv 1 \bmod \Phi(N)$)
Výsledek lze zapsat jako $M^d * c \bmod N$.

Dále je zřejmé, že ze znalosti hodnoty $M^d * c \bmod N$ a hodnoty c lze již snadno vypočítat podpis zprávy M tj. hodnotu $M^d \bmod N$.

K tomu totiž stačí postupně řešit následující modulární rovnice:
$0310 \equiv 105 * M^d \bmod 3337$
$1359 \equiv 105 * M^d \bmod 3337$
$0031 \equiv 105 * M^d \bmod 3337$
$2697 \equiv 105 * M^d \bmod 3337$
....
$2502 \equiv 105 * M^d \bmod 3337$

(Řešení těchto rovnic lze poměrně snadno realizovat i pro velká čísla.)

```
Procedure Equation;
Begin
writeln('Reseni modularni rovnice A=C*X mod N pro ruzna A');
        j:=0; M:=1;
        repeat
                inc(j);
                M1:=C*j-A;
                if M1>0 then
                begin
                M2:=((c*j-A) div N)*N;
                M:=M1-M2;
                end;
        until M=0;
        writeln('A=C*X mod N, X=',j);
end;
```

Vyřešením (viz program Equation) dostaneme následující hodnoty. Označíme je jako posloupnost (**).
1592 0585 1494 3172 644 3080 0647 1855 0707 1740

Posloupnost (**) je Bobův podpis zprávy M = DLUH JE 10 USD
(viz. posloupnost * z úvodu přednášky).

I.3 Závěr

Aby v praxi takovéto problémy nenastávaly a mohly jsme jim předcházet – potřebujeme upravit použití toho, čemu říkáme podpisové schéma (nebo digitální podpis). Zatím je to jen algoritmus. Dané postupy nemají žádnou právní váhu. Chceme-li takovéto postupy v digitálním světě používat - potřebujeme standardy (kvůli kompatibilitě), nezávislé hodnocení bezpečnosti vybraných postupů (kvůli bezpečnosti), potřebujeme upravit chování jednotlivých subjektů (kvůli právní akceptaci) – **POTŘEBUJEME TEDY ZÁKON O ELEKTRONICKÉM PODPISU.**