

# Útoky na digitálne podpisy využitím MD5 kolízií

Ondrej Mikle-Barát  
ondrej.mikle@gmail.com

# Vývoj udalostí (obsah)

- na začiatku nebolo nič\*
  - potom prišli Číňania a priniesli nám dva páry kolidujúcich MD5 vektorov
  - a po pár mesiacoch relatívneho ticha sa s útokmi roztrhlo vrece (Mikle, Kaminsky, {Lenstra, Wang, Weber}...) – binárne formáty, digitálne certifikáty, textové dokumenty, ...
  - teoretické objavy (Hawkes et al., Klíma)
- \*Dobbertin by sa hneval bez tej hviezdičky

# Softwarové balíky a integrita (1)

- MD5 sa často (a dodnes) používa k overeniu integrity
- pre distribuované súbory sa zverejnia MD5 hashe
- typicky sú distribuované vo formátoch exe, zip, tar.bz2, tar.gz, rar, ISO CD-ROM image...
- prípadne miesto MD5 hašu je zverejnený digitálny podpis (zvyčajne gpg/pgp)
- vieme, že sa podpisuje hash, takže dva súbory s rovnakým hashom budú mať rovnaký podpis

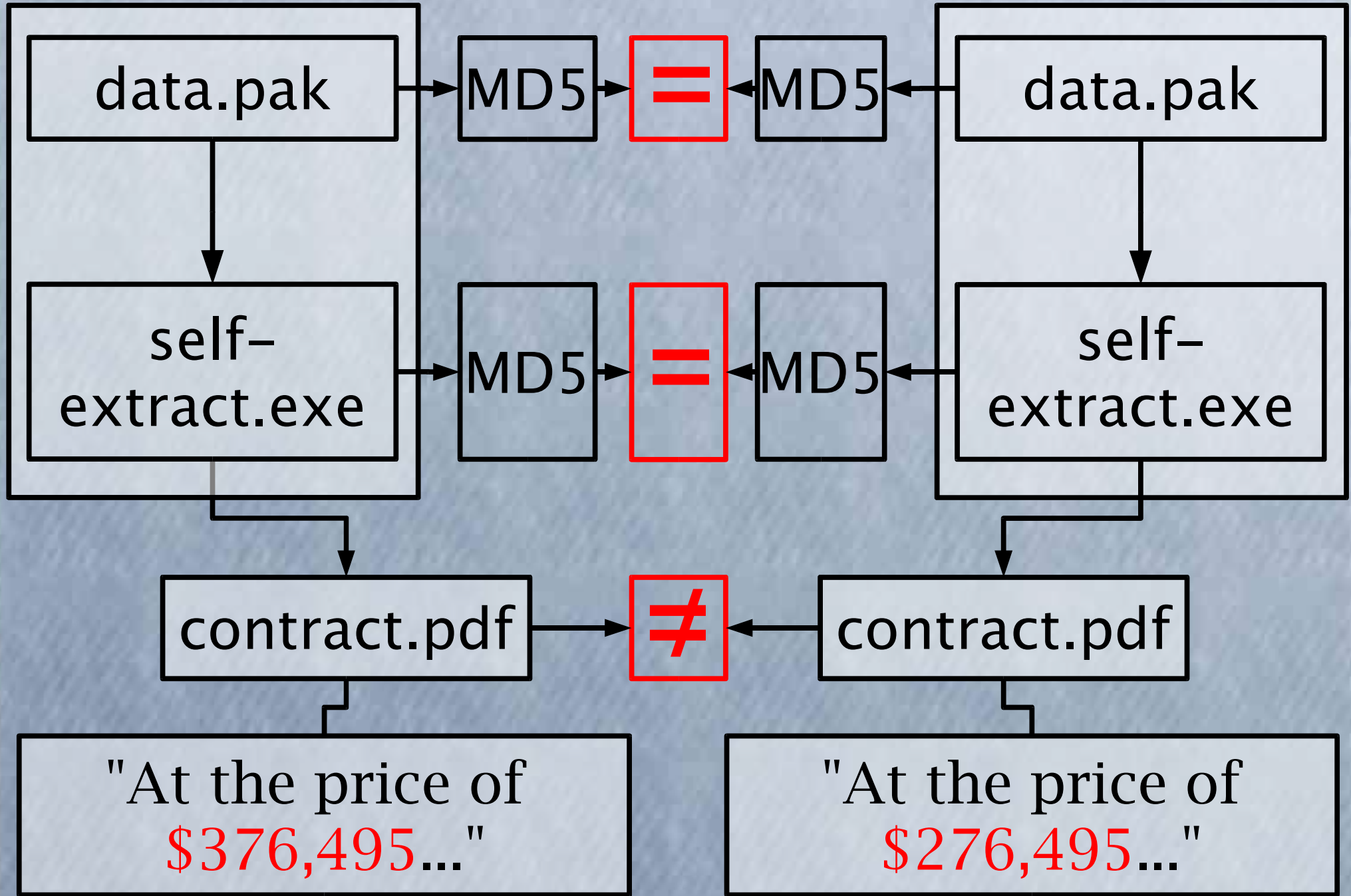
# Softwarové balíky a integrita (2)

Príklad možného typu útoku:

Eva naprogramuje software. Zverejní zdrojové kódy a ich MD5 hashe. Bob, známy kryptoanalytik, ich prezrie a potvrdí, že všetky parametre a implementácia je bezpečná. Eva ale schválne vymenila software za druhú verziu balíku so zadnými vrátkami. Alica, ktorá si Evin software stiahla a verí Bobovi a integrite MD5 hashov, si práve zanesla k sebe zadné vrátka.

archív 1

archív 2



"At the price of  
**\$376,495...**"

"At the price of  
**\$276,495...**"

# Ako taký pár balíkov vytvoriť?

data.pak, verzia 1

```
d131dd02c5e6eec4  
693d9a0698aff95c  
2fcab58712467eab  
4004583eb8fb7...
```

self-extract, pseudokód

```
byte decision;  
  
datafile = open("data.pak");  
seek(data, COLLISION_OFFSET);  
decision = read(datafile);
```

data.pak, verzia 2

```
d131dd02c5e6eec4  
693d9a0698aff95c  
2fcab50712467eab  
4004583eb8fb7...
```

```
if (decision == 0x87)  
    do_good_thing() else  
    do_bad_thing();
```

```
//Na toto stačí jediná  
//známa kolízia!
```

# Konštrukcia kolidujúcich archívov

-v oboch archívoch je binárka self-extract rovnaká (preto bude mať nutne aj rovnaký hash)

-oba data.pak začínajú kolidujúcim vektorom, takže je jedno, čo nalepíme na koniec oboch, kým to bude rovnaký reťazec - a budú mať MD5 hash rovnaký (dôsledok iteratívneho výpočtu hashu)

# Jediná binárka

-je samozrejme možné vytvoriť ju, ale na to sme museli počkať, kým budeme poznať algoritmus výpočtu kolízie pre ľubovoľný inicializačný vektor

-bud' bude program čítať sám seba, alebo ho navrhne tak, že v dátovom segmente si vyhradíme nejaké miesto, kam po skompilovaní umiestnime dopočítanú kolíziu

# PDF

- kolízie pre programy neboli spomínané náhodne
- PDF je v skutočnosti programovací jazyk (aj keď trochu oklieštený)
- je možnosť vkladať ľubovoľné binárne dáta
- má "if-else" konštrukciu, funkcie
- dokonca aj JavaScript tam je
- kolíziou sa dá docieľiť napr. že v jednej verzii sa pár núl navyše vykreslí a v druhej tiež...ale bielou na bielom

# Rôzne Officoidné formáty

- majú makrá (opäť programovací jazyk)
- ale stačí priamo prepísať dátovú časť, aby sme dosiahli viditeľnú zmenu
- na dosiahnutie správnych hodnôt môžeme tie divné zakryť (biele na bielom ;-)) a do políčka, ktoré nás zaujíma, dáme vzorec:  
**divná konštanta z kolízie + bulharská konštanta = výsledok, ktorý chceme**

# And the moral of the story is...

- hoci kolidujúci pár je "binárny bordel", v každom dostatočne zložitom formáte je možné ho niekam upratať tak, aby ovplyvnil to, čo chceme a nepokazil to, čo nechceme
- MD5 je možné použiť už len na test integrity voči prenosovým chybám, ale nie voči zámerným modifikáciám

# Bonus: odolnosť SHA-1 (odhad)

- posledný rok hašovacím funkciám vôbec neprial
- pre nájdenie kolidujúcich vektorov pre SHA-1 treba rádovo  $2^{66}$  operácií
- ako sa dalo očakávať, strhla sa debata, či je to uskutočniteľné dnes (čoskoro) v rozumnom čase a či napr. NSA má na to výpočetnú silu

# DES cracking machine

-odhad z 1993: osoba disponujúca miliónom dolárov dokáže nájsť kľúč k DES za 3 hodiny hrubou silou (tj. vyskúšaním cca. polovice možných kľúčov= $2^{55}$ ) - Wiener: "Efficient DES Key Search"

-Moorov zákon: počet tranzistorov na čipe sa každých cca 18 mesiacov zdvojnásobí

-(2005-1993)/1.5 = 8, tj  $2^8$ -násobné zrýchlenie  
⇒  $2^{55} * 2^8 = 2^{63}$  operácií za 3 hodiny,  $2^{66}$  operácií za 1 deň, lenže...

# DES cracking machine vs SHA-1

- a) operácia SHA-1 nie je úplne zrovnateľná s operáciou DES, SHA-1 je pomalšia
- b) Wiener použil špeciálny FPGA počítač (modifikovaná telefónna ústredňa ;-)), ktorý robí dobre bitové operácie, ale napr. je pomalý na násobenie, atď. - otázka je, či by FPGA pomohlo urýchliť hľadanie kolízií SHA-1
- c) nie je úplne jasné, či by Moorov zákon platí pre FPGA, tj. či sa počítač tohoto druhu natoľko zlepšil

# Pár meraní, DES vs SHA-1

## OpenSSL benchmark (kiB/s), Athlon XP2500+

type	16 bytes	64 bytes	256 bytes
sha1	12638.65k	40327.68k	99057.12k
des cbc	42481.42k	44992.54k	45677.75k

- červeným sú najhoršie výsledky
- kým na bežnom, nešpecializovanom procesore
- nie sú rozdiely až také veľké, na FPGA môžu byť oveľa väčšie

# Záver - odhad odolnosti SHA-1

- odhad 1-denného výpočtu kolízie SHA-1 je momentálne zrejme nereálny
- napriek tomu je ale jasné, že nebude dlho trvať a kolízia bude na svete
- je dosť pravdepodobné, že zložitosť sa ešte zníži nejakým teoretickým objavom a SHA-1 čoskoro postihne rovnaký osud ako MD5
- minútu ticha pre MD5, ďalšiu pre SHA-1 a roky práce všade to povymieňať

# Odkazy & referencie

- Mikle: Practical Attacks on Digital Signatures Using MD5 Message Digest, Cryptology ePrint Archive 2004, <http://eprint.iacr.org/2004/356>
- Kaminsky: MD5 To Be Considered Harmful Someday, Cryptology ePrint Archive 2004, <http://eprint.iacr.org/2004/357>
- Klíma: Nalézání kolizí MD5 - hračka pro notebook, [http://cryptography.hyperlink.cz/md5/MD5\\_kolize.pdf](http://cryptography.hyperlink.cz/md5/MD5_kolize.pdf)
- Lenstra, Wang, Weber: Colliding X.509 Certificates, Cryptology ePrint Archive 2005, <http://eprint.iacr.org/2005/067>