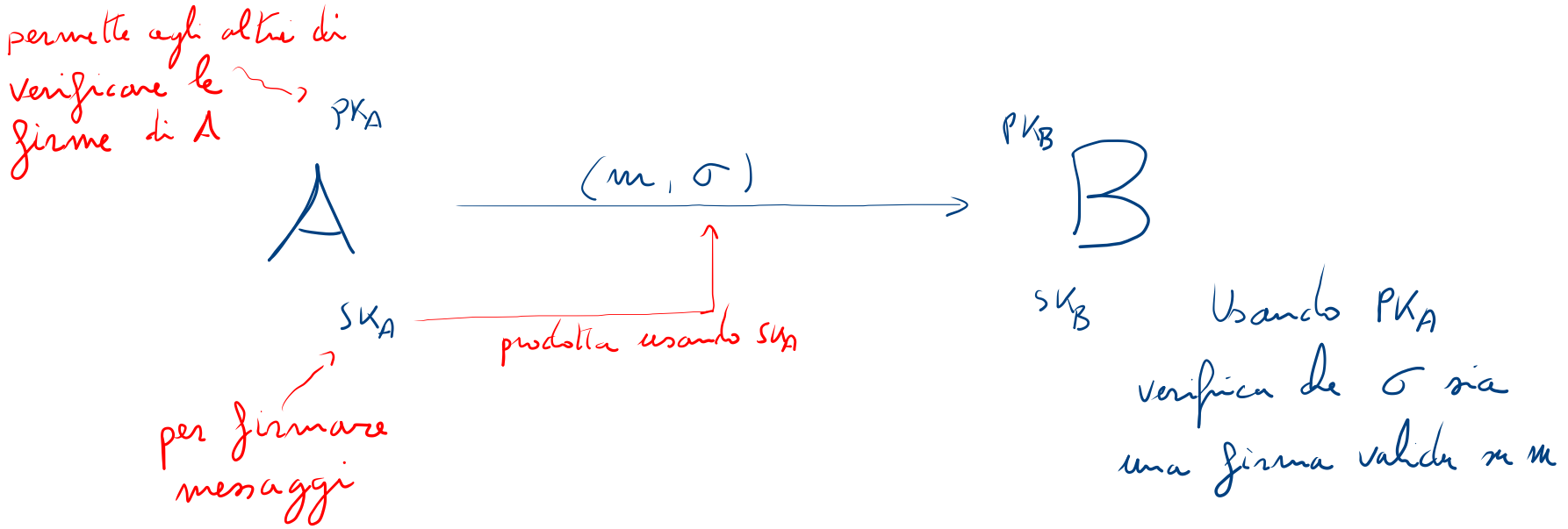
Firme digitali

Schemi di cifratura a chiave pubblica $\Rightarrow$ Confidenzialità nello scenario simmetrico

Schemi di firma digitale $\Rightarrow$ Autenticità ed integrità

↓

Possono essere visti come l'analogo dei codici per l'autenticazione dei messaggi nel contesto simmetrico.



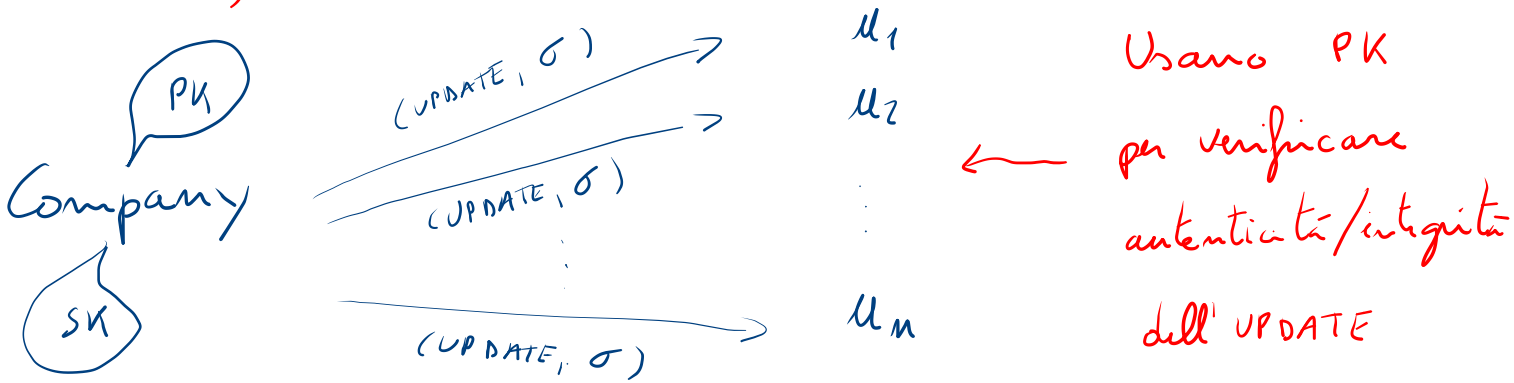
Il proprietario della chiave pubblica PK_A agisce come il mittente. B, o chiunque altro, può verificare la validità di σ su m usando PK_A.

Assunzione: le chiavi pubbliche degli utenti sono autenticate

Non ci sono cioè dubbi sull'associazione chiavi - utenti.

Applicazioni ? Software update

(una pletora ...)



Differenze tra firme e MAC

- Con le firme non occorre un MAC diverso per ogni ricevente
- Pubblicamente verificabili
- Trasferibili $A \xrightarrow{(m,\sigma)} B \xrightarrow[\text{trasferisce}]{(m,\sigma)} C$ Verifica σ su m
- Non ripudiabili. Una volta che A ha firmato un documento, non può successivamente negare.

- Le firme digitali rispetto ai MAC sono più lunghe e 2 o 3 ordini di grandezza più lente da generare / verificare

Osservazione: le firme digitali vengono spesso viste come l'inverso di uno schema di cifratura a chiave pubblica

Storicamente è stato suggerito di:

- firmare, "decifrando" m con SK
- verificare, "cifrando" σ con PK e controllando che coincida con m.

<=> uso delle chiavi invertito

Purtroppo questo approccio in generale non ha fondamento.

In molti casi non è applicabile.

Quando lo è, le costruzioni ottenute non sempre sono sicure.

Definizione formale

DEFINITION 12.1 *A* (digital) signature scheme *consists of three probabilistic polynomial-time algorithms* (Gen, Sign, Vrfy) *such that:*

1. *The* key-generation algorithm Gen *takes as input a security parameter* 1^n *and outputs a pair of keys* (pk, sk). *These are called the* public key *and the* private key, *respectively. We assume that* pk *and* sk *each has length at least* n, *and that* n *can be determined from* pk *or* sk.

2. *The* signing algorithm Sign *takes as input a private key* sk *and a message* m *from some message space (that may depend on* pk*). It outputs a signature* σ, *and we write this as* $\sigma \leftarrow \mathsf{Sign}_{sk}(m)$.

3. *The deterministic* verification algorithm Vrfy *takes as input a public key* pk, *a message* m, *and a signature* σ. *It outputs a bit* b, *with* $b = 1$ *meaning* valid *and* $b = 0$ *meaning* invalid. *We write this as* $b := \mathsf{Vrfy}_{pk}(m, \sigma)$.

It is required that except with negligible probability over (pk, sk) *output by* $\mathsf{Gen}(1^n)$, *it holds that* $\mathsf{Vrfy}_{pk}(m, \mathsf{Sign}_{sk}(m)) = 1$ *for every (legal) message* m.

If there is a function ℓ *such that for every* (pk, sk) *output by* $\mathsf{Gen}(1^n)$ *the message space is* $\{0,1\}^{\ell(n)}$, *then we say that* (Gen, Sign, Vrfy) *is a* signature scheme for messages of length $\ell(n)$.

Sicurezza di uno schema di firma digitale

The signature experiment $\mathsf{Sig\text{-}forge}_{\mathcal{A},\Pi}(n)$**:**

1. $\mathsf{Gen}(1^n)$ *is run to obtain keys* (pk, sk).
2. *Adversary* $\mathcal{A}$ *is given* pk *and access to an oracle* $\mathsf{Sign}_{sk}(\cdot)$. *The adversary then outputs* (m, σ). *Let* $\mathcal{Q}$ *denote the set of all queries that* $\mathcal{A}$ *asked its oracle.*
3. $\mathcal{A}$ succeeds *if and only if (1)* $\mathsf{Vrfy}_{pk}(m, \sigma) = 1$ *and (2)* $m \notin \mathcal{Q}$. *In this case the output of the experiment is defined to be* 1.

DEFINITION 12.2 *A signature scheme* $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ *is* existentially unforgeable under an adaptive chosen-message attack, *or just* secure, *if for all probabilistic polynomial-time adversaries* $\mathcal{A}$, *there is a negligible function* negl *such that:*

$$\Pr[\mathsf{Sig\text{-}forge}_{\mathcal{A},\Pi}(n) = 1] \leq \mathsf{negl}(n).$$

Approccio ibrido

Come per gli schemi di cifratura a chiave pubblica l'approccio ibrido permette di guadagnare efficienza, così per gli schemi di firma un approccio ibrido permette di ottenere schemi più performanti

Nel caso delle firme questo approccio è costituito dal paradigma hash-and-sign

Procede sulla falsariga dell'approccio hash-and-mac

$m \in \{0,1\}^*$, $H(m) \in \{0,1\}^\ell$, $\mathrm{Sign}_{SK}(H(m))$

Paradigma Hash-and-sign

CONSTRUCTION 12.3

Let $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme for messages of length $\ell(n)$, and let $\Pi_H = (\mathsf{Gen}_H, H)$ be a hash function with output length $\ell(n)$. Construct signature scheme $\Pi' = (\mathsf{Gen}', \mathsf{Sign}', \mathsf{Vrfy}')$ as follows:

- Gen': on input 1^n, run $\mathsf{Gen}(1^n)$ to obtain (pk, sk) and run $\mathsf{Gen}_H(1^n)$ to obtain s; the public key is $\langle pk, s\rangle$ and the private key is $\langle sk, s\rangle$.
- Sign': on input a private key $\langle sk, s\rangle$ and a message $m \in \{0,1\}^*$, output $\sigma \leftarrow \mathsf{Sign}_{sk}(H^s(m))$.
- Vrfy': on input a public key $\langle pk, s\rangle$, a message $m \in \{0,1\}^*$, and a signature σ, output 1 if and only if $\mathsf{Vrfy}_{pk}(H^s(m), \sigma) \stackrel{?}{=} 1$.

The hash-and-sign paradigm.

Sicurezza della costruzione

Π + Π_H $\Rightarrow$ hash-and-sign

Π sicuro per firma di $m \in \{0,1\}^{\ell}$

Π_H hash resistente a collisioni

hash-and-sign sicuro per $m \in \{0,1\}^*$

Teorema Se Π è uno schema di firma digitale per messaggi di lunghezza ℓ sicuro e Π_H è una funzione hash resistente a collisioni, allora la costruzione hash-and-sign è sicura per messaggi di lunghezza arbitraria

Costruzioni

CONSTRUCTION 12.5

Let GenRSA be as in the text. Define a signature scheme as follows:

- Gen: on input 1^n run GenRSA(1^n) to obtain (N, e, d). The public key is $\langle N, e \rangle$ and the private key is $\langle N, d \rangle$.
- Sign: on input a private key $sk = \langle N, d \rangle$ and a message $m \in \mathbb{Z}_N^*$, compute the signature

$$\sigma := [m^d \bmod N].$$

- Vrfy: on input a public key $pk = \langle N, e \rangle$, a message $m \in \mathbb{Z}_N^*$, and a signature $\sigma \in \mathbb{Z}_N^*$, output 1 if and only if

$$m \stackrel{?}{=} [\sigma^e \bmod N].$$

The plain RSA signature scheme.

È facile verificare che, per firme generate correttamente, risulta

$$\sigma^e = (m^d)^e = m^{[e \cdot d \bmod \varphi(N)]} = m^1 = m \bmod N$$

Circa la sicurezza, poiché il problema RSA è ritenuto difficile, si potrebbe pensare che sia automaticamente difficile produrre contraffazioni. Si noti che

Assunzione RSA $\Longrightarrow$ difficile produrre contraffazioni di messaggi m scelti uniformemente

Non assicura nulla sulla difficoltà di produrre firme di messaggi non uniformi (o scelti da Adv)

Un attacco senza messaggi

Adv usa soltanto (N, e).

Data (N, e), sceglie $\sigma \in \mathbb{Z}_N^*$, calcola

$m = [\sigma^e \bmod N]$ e dà in output (m, σ)!

Ovviamente è una contraffazione. La firma σ è valida su m e non è stata mai generata dal legittimo firmatario!

Adv non ha controllo su m, ma è sufficiente a dimostrare che la costruzione non soddisfa la definizione.

Adv, inoltre, potrebbe avere controllo "parziale"

Contraffazione di un messaggio arbitrario

Adv riceve due firme dal firmante. Produce una contraffazione su m di sua scelta.

Adv: sceglie $m_1, m_2 \in \mathbb{Z}_N^*$ tali che $m = m_1 \cdot m_2 \bmod N$

ottiene σ_1 e σ_2, firme di m_1 ed m_2

Dà in output $\sigma = [\sigma_1 \cdot \sigma_2 \bmod N]$ su m

Risulta:

$$\sigma^e = (\sigma_1 \cdot \sigma_2)^e = (m_1^d \cdot m_2^d)^e = (m_1^{cd} \cdot m_2^{ed}) = m_1 \cdot m_2$$

$$= m \bmod N$$

Nota: l'attacco può essere generalizzato.

Dato un insieme di q firme su $M = \{m_1, \dots, m_q\}$,

Adv può produrre firme su $2^q - q$ altri

messaggi. Devastante!

RSA - FDH (Full domain hash)

(PKCS #1 v2.1)

CONSTRUCTION 12.6

Let GenRSA be as in the previous sections, and construct a signature scheme as follows:

- Gen: on input 1^n, run GenRSA(1^n) to compute (N, e, d). The public key is $\langle N, e\rangle$ and the private key is $\langle N, d\rangle$.

  As part of key generation, a function $H : \{0,1\}^* \to \mathbb{Z}_N^*$ is specified, but we leave this implicit.

- Sign: on input a private key $\langle N, d\rangle$ and a message $m \in \{0,1\}^*$, compute

$$\sigma := [H(m)^d \bmod N].$$

- Vrfy: on input a public key $\langle N, e\rangle$, a message m, and a signature σ, output 1 if and only if $\sigma^e \stackrel{?}{=} H(m) \bmod N$.

The RSA-FDH signature scheme.

Quali proprietà deve avere H?

- Sicuramente deve essere difficile da invertire (primo attacco)
- Non deve ammettere "relazioni moltiplicative" (secondo attacco)

e.g., $H(m) = H(m_1) \cdot H(m_2) \bmod N$

- Deve essere difficile trovare collisioni

i.e., $m_1 \neq m_2$ tali che $H(m_1) = H(m_2)$

Se modelliamo H come un oracolo casuale sono soddisfatte tutte le condizioni.

Teorema: RSA difficile rel. a GenRSA + H ROM ⇒ RSA-FDH sicuro

Sketch: se esistesse A che produce contraffazioni con probabilità non trascurabili

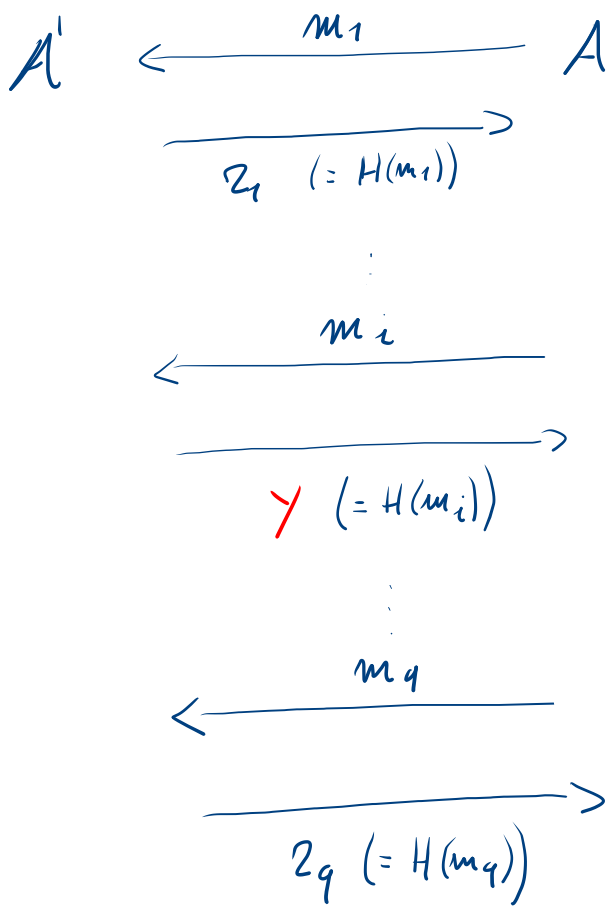
⇒ esisterebbe A' che risolve il problema RSA con probabilità non trascurabile

A' simula l'esperimento $Sig\text{-}forge_{A,H}(n)$ per A e risponde alle sue query.

Caso semplice: A invia solo query ad H (no firme)

A' risponde alle query come segue: → poly(n)

su input (N, e, y), risponde alle q query di A scegliendo valori casuali in $\mathbb{Z}_N^*$, tranne in un caso, scelto unif. a caso



Simula $\text{Sign-forge}_{A,\Pi}^{(n)}$ per A

i-esima query $(i \in \{1, \dots, q\})$

Poiché H è un oracolo casuale i valori H(m) sono uniformi

Visto che A' sceglie gli z_j, con $j \neq i$, uniformi, A non nota alcuna differenza nella distribuzione dei valori che riceve.

Circa y, poiché x è scelto uniformemente nell'esperimento che definisce il problema RSA, anch'esso è uniforme (vi ricordo che RSA è una permutazione).

Quindi, se A produce una contraffazione (m, σ) su un m di cui ha chiesto H(m), con probabilità 1/q questo m è m_i!

Pertanto, se A' dà in output σ come soluzione al problema RSA, risulta

$$\sigma^e = H(m) = H(m_i) = y \bmod N$$

con probabilità

$$1/q \cdot \underbrace{non\text{-}negl(n)}_{\text{probabilità con cui } A \text{ genera una contraffazione } (m,\sigma)}$$

Pertanto, A' è efficiente se A è efficiente e risolve il problema RSA con probabilità non trascurabile!

Problema: e se invece A chiede anche firme di messaggi di sua scelta all'oracolo di firma prima di produrre la contraffazione?

A non conosce d per firmare messaggi!

Come fa a simulare l'oracolo di firma?

Idea: A, per m_j, sceglie un valore casuale σ_j e calcola $H(m_j) = \sigma_j^e \bmod N$

Nota che σ_j uniforme $\Longrightarrow$ $H(m_j) = \sigma_j^e \bmod N$ uniforme (RSA è una permutazione)

Quindi A', alla prima query per m_j, sia essa per l'oracolo H che per l'oracolo $Sign$, genera e memorizza la tripla

$$\left(m_j \,,\ \sigma_j \,,\ \sigma_j^{\,e} \right)$$

m_j : m — σ_j : $H(m)^d$ — $\sigma_j^{\,e}$: $H(m)$

← le risposte alle hash query sono uniformemente distribuite come in $Sign\text{-}forge_{A,\Pi}(n)$

Con questa strategia A' è in grado di rispondere consistentemente ad entrambi i tipi di query

Nota: stiamo "programmando" l'oracolo (3ª proprietà ROM)

RSA PKCS #1 v2.1

Lo standard include uno schema di firma che è una variante di RSA-FDH

- la firma dipende da un valore casuale (detto SALT) scelto dal firmante all'atto della firma

Osservazione: ai fini della sicurezza di RSA-FDH è importante che il codominio di H sia molto prossimo a tutto $\mathbb{Z}_N^*$. Non è sufficiente usare SHA-2 o SAA-3. Occorre costruire H attraverso applicazioni ripetute.

Probabilistic Signature Scheme

(RSA-FDH randomizzato)

Ogni messaggio può avere più firme

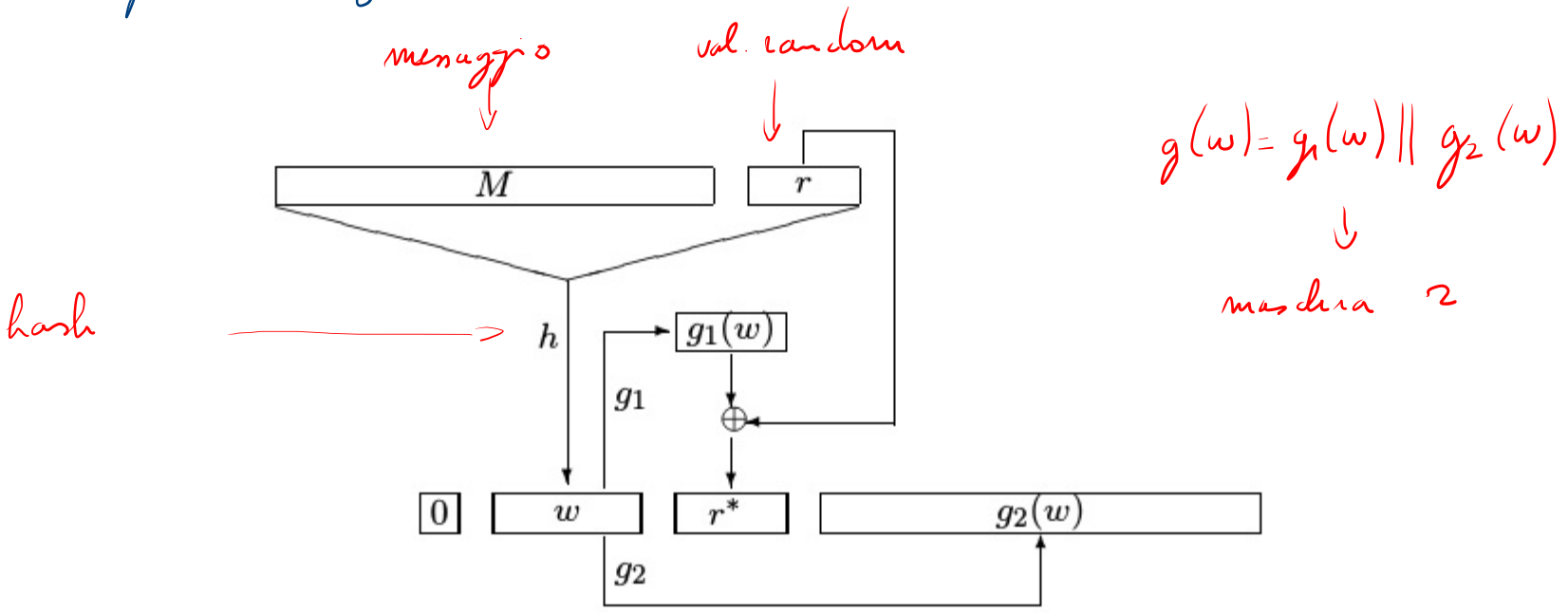


Figure 1: PSS: Components of image $y = 0 \| w \| r^* \| g_2(w)$ are darkened. The signature of M is $y^d \bmod N$.

Algoritmi di firma e verifica

SignPSS (M)

$r \xleftarrow{R} \{0,1\}^{k_0}$; $w \leftarrow h(M \parallel r)$; $r^* \leftarrow g_1(w) \oplus r$

$y \leftarrow 0 \parallel w \parallel r^* \parallel g_2(w)$

return $y^d \bmod N$

VerifyPSS (M, x)

$y \leftarrow x^e \bmod N$

Break up y as $b \parallel w \parallel r^* \parallel \gamma$. (That is, let b be the first bit of y, w the next k_1 bits, r^* the next k_0 bits, and γ the remaining bits.)

$r \leftarrow r^* \oplus g_1(w)$

if ($h(M \parallel r) = w$ and $g_2(w) = \gamma$ and $b = 0$) **then return** 1

else return 0

Perché PSS ?

- la riduzione di sicurezza per RSA-FDH non è stretta

Informalmente: se A rompe Π in tempo t e prob ε

$\Rightarrow$ A' risolve X in tempo $\approx t$ e prob $\approx \varepsilon$

la riduzione è stretta

Nella pratica è importante:

- in base alle nostre conoscenze attuali del momento, se sappiamo che X richiede tempo t per una prob di successo ε, allora ogni A di tempo t rompe Π con prob di successo al più ε.

Ragioniamo su un esempio

N modulo di K bit

Unici modi per violare RSA passano per la fattorizzazione

Conoscenza crittografica: miglior alg. di fattorizzazione richiede

$$T \approx e^{K/4} \quad (\text{prob. } 1)$$

(ho già semplificato ma ... semplifichiamo ancora, $2^{K/4}$)

Posso usare tale assunzione per stimare la prob di successo

di A di tempo $t(K)$ $\Rightarrow$ $\varepsilon(K) = \dfrac{t(K)}{2^{K/4}}$

Se voglio ammettere prob di successo 2^{-60}, allora

$$t(K) = 2^{K/5} \cdot 2^{-60} = 2^{\frac{K-250}{5}}$$

Se il mio modulo N ha $K = 1024$ bit, un Alg di tempo al più $2^{\frac{1024-250}{5}} = 2^{196}$ ha al più prob 2^{-60} di fattorizzare N.

Assunzione: $RSA \in (t, \varepsilon) = (2^{196}, 2^{-60})$-sicuro

Riduzione RSA-FDH: RSA (t', ε')-sicuro (hash query, sign query)

$\Rightarrow$ RSA-FDH (t, ε)-sicuro dove:

$$\begin{cases} t(K) = t'(K) - [q_h + q_s] \cdot K^3 \\ \varepsilon(K) = [q_h + q_s] \cdot \varepsilon'(K) \end{cases}$$

Con i nostri parametri, ammettendo $q_h = 2^{60}$ e $q_s = 2^{30}$,

$$\ell(K) = 2^{196} - [2^{60} + 2^{30}] \cdot K^3$$

$$\geq 2^{196} - 2^{60} \cdot (1024)^3 \qquad \left(2^{K/5} - 2^{60} \cdot K^3\right) \quad \text{OK!}$$

(tempo ancora relativamente alto ammesso per gli A.lv)

$$\varepsilon(K) = [2^{60} + 2^{30}] \cdot 2^{-60}$$

$$\geq 2^{60} \cdot 2^{-60} = 1 \quad !$$

Per rendere $\varepsilon(K)$ piccolo, i.e., $\varepsilon(K) = 2^{-60}$

devo scegliere K più grande. Precisamente,

procedendo a ritroso ...

$$2^{-50} = 2^{60} \cdot \varepsilon'(K) \quad \Rightarrow \quad \varepsilon'(K) = \frac{2^{-50}}{2^{60}} = 2^{-100}$$

Essendo $\varepsilon(K) = \frac{t'(K)}{2^{K/4}}$ e volendo, per esempio, mantenere lo stesso limite di tempo per gli Adv che cercano di fattorizzare, deve essere

$$t'(K) = 2^{-100} \cdot 2^{K/4} \iff 2^{196} = 2^{-100} \cdot 2^{K/4}$$

$$\iff 2^{K/4} = 2^{196+100} = 2^{296} \iff K = 4 \cdot 296$$

$$\iff K = 1184 \text{ bit.}$$

La taglia K del modulo passa da 1024 a 1184 bit

RSA-FDH e PSS

- la riduzione del problema RSA a RSA-FDH non è stretta

- PSS è stato introdotto con l'obiettivo di ottenere una riduzione stretta

  Nota: progettazione orientata all'ottenimento di una riduzione migliore (trend nella ricerca ...)

- Una riduzione migliore per RSA-FDH è stata prodotta qualche anno dopo

Per i curiosi ...

RSA - FDH e PSS

The **Exact** Security of Digital Signatures— How to Sign with **RSA** and **Rabin**

Mihir Bellare* Phillip Rogaway†

March 14, 1996

Abstract

We describe an RSA-based signing scheme called PSS which combines essentially optimal efficiency with attractive security properties. Signing takes one RSA decryption plus some hashing, verification takes one RSA encryption plus some hashing, and the size of the signature is the size of the modulus. Assuming the underlying hash functions are ideal, our schemes are not only provably secure, but are so in a *tight* way— an ability to forge signatures with a certain amount of computational resources implies the ability to invert RSA (on the same size modulus) with about the same computational effort. Furthermore, we provide a second scheme which maintains all of the above features and in addition provides message recovery. These ideas extend to provide schemes for Rabin signatures with analogous properties; in particular their security can be tightly related to the hardness of factoring.

Per i curiosi ...

RSA - FDH

On the exact security of Full Domain Hash

Jean-Sébastien Coron

Ecole Normale Supérieure
45 rue d'Ulm
Paris, F-75230, France
coron@clipper.ens.fr

Gemplus Card International
34 rue Guynemer
Issy-les-Moulineaux, F-92447, France
jean-sebastien.coron@gemplus.com

Abstract. The Full Domain Hash (FDH) scheme is a RSA-based signature scheme in which the message is hashed onto the full domain of the RSA function. The FDH scheme is provably secure in the random oracle model, assuming that inverting RSA is hard. In this paper we exhibit a slightly different proof which provides a tighter security reduction. This in turn improves the efficiency of the scheme since smaller RSA moduli can be used for the same level of security. The same method can be used to obtain a tighter security reduction for Rabin signature scheme, Paillier signature scheme, and the Gennaro-Halevi-Rabin signature scheme.