Schemi di firma basati sul problema DL

Schema di firme di Schnorr

Per comprenderne la strategia di progettazione discutiamo prima di schemi di identificazione a chiave pubblica

Schema di identificazione: protocollo interattivo che permette ad una parte di provare la propria identità (i.e., autenticare se stesso) ad un'altra parte.

$$
\begin{tikzcd}
P \arrow[r, shift left=2] \arrow[r, leftarrow] \arrow[r, shift right=2] & V
\end{tikzcd}
$$

(prover) (verifier)

P e V non condividono alcuna informazione a-priori

Consideriamo solo protocolli di identificazione in tre passi

$P \longrightarrow$ rappresentato da P_1, P_2
(due algoritmi)

$V \longrightarrow$ rappresentato da un solo alg. V

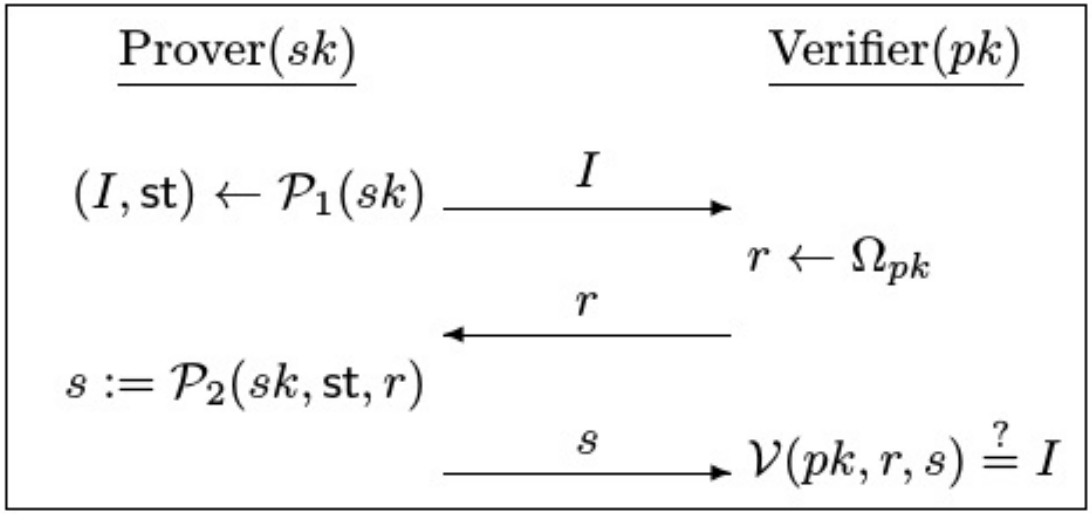

FIGURE 12.1: A three-round identification scheme.

Si richiede che, se il provatore è legittimo, il verificatore lo identifichi sempre. Consideriamo solo schemi non degeneri: per ogni chiave privata sk ed ogni messaggio I iniziale la probabilità che $\mathcal{P}_1(sk)$ dia I è trascurabile.

Sicurezza ?

Ovviamente Adv, senza disporre di sk, non deve essere in grado di farsi identificare da V ...

(i.e., impersonare P ...)

... anche se riesce ad "ascoltare" diverse esecuzioni del protocollo tra P e V.

Formalizzazione: Sia Trans_{sk} un oracolo che, invocato senza input, esegue il protocollo tra due parti e restituisce ad Adv la trascrizione dei messaggi (transcript) che si sono scambiati, ovvero (I, r, s)

Sia $\Pi = (Gen, P_1, P_2, V)$ uno schema di identificazione.

A un avversario ppt.

The identification experiment $\mathsf{Ident}_{\mathcal{A},\Pi}(n)$:

1. $\mathsf{Gen}(1^n)$ *is run to obtain keys* (pk, sk).
2. *Adversary* $\mathcal{A}$ *is given* pk *and access to an oracle* Trans_{sk} *that it can query as often as it likes.*
3. *At any point during the experiment,* $\mathcal{A}$ *outputs a message* I. *A uniform challenge* $r \in \Omega_{pk}$ *is chosen and given to* $\mathcal{A}$*, who responds with some* s. *(*$\mathcal{A}$ *may continue to query* Trans_{sk} *even after receiving* r*.)*
4. *The experiment outputs* 1 *if and only if* $\mathcal{V}(pk, r, s) \stackrel{?}{=} I$.

DEFINITION 12.8 *An identification scheme* $\Pi = (\mathsf{Gen}, \mathcal{P}_1, \mathcal{P}_2, \mathcal{V})$ *is* secure against a passive attack, *or just* secure, *if for all probabilistic polynomial-time adversaries* $\mathcal{A}$, *there exists a negligible function* negl *such that:*

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

Ovviamente possono essere considerati e modellati anche Adv attivi. Ma non è richiesto per produrre uno schema di firma digitale.

La trasformazione di Fiat e Shamir offre un metodo per convertire uno schema di identificazione interattivo in uno schema di firma non interattivo.

Idea: il prover esegue il protocollo di identificazione da solo, rimuovendo l'interazione usando una funzione hash.

CONSTRUCTION 12.9

Let $(\mathsf{Gen}_{\mathsf{id}}, \mathcal{P}_1, \mathcal{P}_2, \mathcal{V})$ be an identification scheme, and construct a signature scheme as follows:

- Gen: on input 1^n, simply run $\mathsf{Gen}_{\mathsf{id}}(1^n)$ to obtain keys pk, sk.

  The public key pk specifies a set of challenges Ω_{pk}. As part of key generation, a function $H : \{0,1\}^* \to \Omega_{pk}$ is specified, but we leave this implicit.

- Sign: on input a private key sk and a message $m \in \{0,1\}^*$, do:

  1. Compute $(I, \mathsf{st}) \leftarrow \mathcal{P}_1(sk)$.
  2. Compute $r := H(I, m)$.
  3. Compute $s := \mathcal{P}_2(sk, \mathsf{st}, r)$.

  Output the signature (r, s).

- Vrfy: on input a public key pk, a message m, and a signature (r, s), compute $I := \mathcal{V}(pk, r, s)$ and output 1 if and only if $H(I, m) \stackrel{?}{=} r$.

The Fiat–Shamir transform.

Una firma (r, s) è legata ad un messaggio specifico m perché r è una funzione sia di I che di m.

Cambiando $m \longrightarrow$ cambia totalmente r

Se H viene modellata come un oracolo casuale che mappa gli input uniformemente su Ω_{pk}, allora r è uniformemente distribuito..

Discende che per ...

Adv è tanto difficile trovare una firma valida (r, s) su un messaggio m quanto lo sarebbe impersonare il prover in una esecuzione onesta del protocollo di identificazione.

Teorema. Π schema di identificazione.

Π' schema di firme ottenuto da Π

applicando la trasformazione di Fiat e Shamir

Se Π è sicuro e H è ROM

$\Rightarrow$ Π' è sicuro

Schema di identificazione di Schnorr

Il prover esegue $\mathcal{G}(1^n) \Rightarrow (\mathbb{G}, q, g)$, sceglie $x \in \mathbb{Z}_q$ uniformemente e pone $y = g^x$

$pk = (\mathbb{G}, q, g, y)$ (chiave pubblica)

$sk = (\mathbb{G}, q, g, x)$ (chiave privata)

Prover(x)		Verifier$(\mathbb{G}, q, g, y)$
$k \leftarrow \mathbb{Z}_q$		
$I := g^k$	$\xrightarrow{\quad I \quad}$	
		$r \leftarrow \mathbb{Z}_q$
	$\xleftarrow{\quad r \quad}$	
$s := [rx + k \bmod q]$		
	$\xrightarrow{\quad s \quad}$	check whether $g^s \cdot y^{-r} \stackrel{?}{=} I$

FIGURE 12.2: An execution of the Schnorr identification scheme.

È corretto perché:

$$g^s \cdot y^{-2} = g^{(2x+k)} \cdot (g^x)^{-2} = g^{2x+k-2x} = g^k = I$$

Perché risulta sicuro?

Prima osservazione: l'ascolto di esecuzioni del protocollo non aiuta Adv.

Infatti, Adv può simulare transcript di esecuzioni oneste del protocollo da solo, sfruttando soltanto la chiave pubblica e non conoscendo la chiave privata.

Come? Invertendo l'ordine dei passi

- prima sceglie indipendentemente ed uniformemente $z, s \in \mathbb{Z}_q$ e poi pone $I = g^s y^{-z}$

Distribuzione reale

(I, z, s)

elemento uniforme di G

elemento uniforme di $\mathbb{Z}_q$, indipendente da I

univocamente determinato come $s = \log_g (I \cdot y^z)$

Distribuzione simulata

(I, z, s)

elementi uniformi di $\mathbb{Z}_q$ ed indipendenti

elemento uniforme di G ed indipendente da z (essendo s indip)

Vale ancora $s = \log_g (I \cdot y^z)$

Distribuzioni identiche!

Possiamo, quindi, ridurci all'analisi di Adv che non ascoltano

chiave pubblica

$$\mathrm{Adv}(y) \xrightarrow{\quad I \quad}$$

$$\xleftarrow{\quad z \quad}$$

$$s\,? \xrightarrow{\quad s \quad} \quad \text{tale che } g^{s} y^{-z} = I$$

Se Adv fosse in grado di calcolare valori di s giusti efficientemente e con alta probabilità, allora sarebbe in grado di calcolare risposte corrette s_1 ed s_2 ad almeno due sfide $z_1, z_2 \in \mathbb{Z}_q$. Nota che:

$$g^{s_1} y^{-z_1} = I = g^{s_2} \cdot y^{-z_2} \Rightarrow g^{s_1 - s_2} = y^{z_1 - z_2}.$$

Ciò implica che Adv può calcolare esplicitamente

$$\log_g y = \left[(s_1 - s_2)(z_1 - z_2)^{-1} \bmod q \right]$$

ovvero il logaritmo discreto di y, contraddicendo la presunta difficoltà del problema DL.

Teorema. Se DL è difficile in G $\Longrightarrow$ lo schema di Schnorr è sicuro

Lo schema di firma di Schnorr si ottiene applicando la trasformazione di Fiat e Shamir

CONSTRUCTION 12.12

Let $\mathcal{G}$ be as described in the text.

- Gen: run $\mathcal{G}(1^n)$ to obtain $(\mathbb{G}, q, g)$. Choose a uniform $x \in \mathbb{Z}_q$ and set $y := g^x$. The private key is x and the public key is $(\mathbb{G}, q, g, y)$. As part of key generation, a function $H : \{0,1\}^* \to \mathbb{Z}_q$ is specified, but we leave this implicit.
- Sign: on input a private key x and a message $m \in \{0,1\}^*$, choose uniform $k \in \mathbb{Z}_q$ and set $I := g^k$. Then compute $r := H(I, m)$, followed by $s := [rx + k \bmod q]$. Output the signature (r, s).
- Vrfy: on input a public key $(\mathbb{G}, q, g, y)$, a message m, and a signature (r, s), compute $I := g^s \cdot y^{-r}$ and output 1 if $H(I, m) \stackrel{?}{=} r$.

The Schnorr signature scheme.

DSA ed ECDSA

Il Digital Signature Algorithm e l' Elliptic Curve Digital Signature Algorithm sono entrambi basati sul problema DL su classi di gruppi differenti.

Stesso template $\longrightarrow$ possono essere visti come costruiti da uno schema di identificazione sottostante

Consideriamo il seguente schema di identificazione

chiave privata $\rightarrow$ x, $y = g^x$

(G, q, g, y)

chiave pubblica

P $\qquad\qquad\qquad\qquad$ V

Sceglie $K \in \mathbb{Z}_q$ unif., calcola $I = g^K$

$\xrightarrow{\quad I \quad}$

Sceglie $\alpha, z \in \mathbb{Z}_q$ unif. e li invia come challenge

$\xleftarrow{\quad \alpha, z \quad}$

Calcola ed invia

$$s = \left[K^{-1} \cdot (\alpha + x z) \bmod q\right]$$

$\xrightarrow{\quad s \quad}$

Accetta se $s \neq 0$ e

$$g^{\alpha s^{-1}} \cdot y^{z \cdot s^{-1}} = I$$

Si noti che risulta $s \neq 0$ a meno che $\alpha = -x_2 \bmod q$ (che accade con probabilità trascurabile)

Pertanto, s^{-1} esiste e risulta: $\left(s = K^{-1} \cdot (\alpha + x_2) \bmod q \right)$

$$g^{\alpha s^{-1}} \cdot y^{2 \cdot s^{-1}} = g^{\alpha \cdot s^{-1}} \cdot \left(g^{x}\right)^{2 \cdot s^{-1}} = g^{(\alpha + x \cdot 2) \cdot s^{-1}}$$

$$= g^{(\alpha + x_2)\, (\alpha + x_2)^{-1} \cdot K} = g^{K} = I$$

Quindi, lo schema è corretto.

Si può dimostrare sicuro se il problema DL è difficile relativamente a $\mathcal{G}(1^n)$.

Sketch della prova

- transcript di esecuzioni oneste possono essere simulati

  scelti unif. $\alpha, z \in \mathbb{Z}_q$, $s \in \mathbb{Z}_q^*$, $I = g^{\alpha s^{-1}} \cdot y^{z \cdot s^{-1}}$

- se Adv da' in output I per cui può dare risposte

  corrette $s_1, s_2 \in \mathbb{Z}_q^*$ a challenge distinte $(\alpha, z_1), (\alpha, z_2)$

  allora $$g^{\alpha \cdot s_1^{-1}} \cdot y^{z_1 \cdot s_1^{-1}} = I = g^{\alpha s_2^{-1}} \cdot y^{z_2 \cdot s_2^{-1}}$$

  $$\Rightarrow g^{\alpha(s_1^{-1} - s_2^{-1})} = y^{z_2 s_2^{-1} - z_1 \cdot s_1^{-1}} \Rightarrow$$ posso calcolare $\log_g y$

Vale anche per challenge $(\alpha_1, z_1), (\alpha_2, z_2)$.

Gli schemi di firma DSA / ECDSA sono costruiti facendo collassare lo schema di identificazione in un algoritmo non-interattivo eseguito dal firmante

Rispetto alla trasformazione di Fiat e Shamir, operiamo come segue:

- $\alpha = H(m)$, m messaggio, H funzione hash
- $r = F(I)$, $F : G \to \mathbb{Z}_q$, F funzione semplice

In DSA, G sottogruppo di ordine q di $\mathbb{Z}_p^*$ p primo

$$F(I) \overset{\text{def}}{=} [I \bmod q]$$

In ECDSA, G sottogruppo di ordine q di una curva ellittica $E(\mathbb{Z}_p)$, p primo

$$F[\underbrace{(x, y)}_{\text{punto di } E(\mathbb{Z}_p)}] \stackrel{\text{def}}{=} [x \bmod q]$$

Entrambi gli schemi possono essere descritti in modo astratto come segue

CONSTRUCTION 12.13

Let $\mathcal{G}$ be as in the text.

- Gen: on input 1^n, run $\mathcal{G}(1^n)$ to obtain $(\mathbb{G}, q, g)$. Choose uniform $x \in \mathbb{Z}_q$ and set $y := g^x$. The public key is $\langle \mathbb{G}, q, g, y \rangle$ and the private key is x.

  As part of key generation, two functions $H : \{0,1\}^* \to \mathbb{Z}_q$ and $F : \mathbb{G} \to \mathbb{Z}_q$ are specified, but we leave this implicit.

- Sign: on input the private key x and a message $m \in \{0,1\}^*$, choose uniform $k \in \mathbb{Z}_q^*$ and set $r := F(g^k)$. Then compute $s := [k^{-1} \cdot (H(m) + xr) \bmod q]$. (If $r = 0$ or $s = 0$ then start again with a fresh choice of k.) Output the signature (r, s).

- Vrfy: on input a public key $\langle \mathbb{G}, q, g, y \rangle$, a message $m \in \{0,1\}^*$, and a signature (r, s) with $r, s \neq 0 \bmod q$, output 1 if and only if

$$r \stackrel{?}{=} F\left(g^{H(m) \cdot s^{-1}} y^{r \cdot s^{-1}}\right).$$

DSA and ECDSA—abstractly.

Se il problema DL è difficile relativamente a $\mathcal{G}(1^n)$ ed H ed F sono modellati come oracoli casuali, allora la costruzione generica è sicura.

Osservazioni. Il valore $K \in \mathbb{Z}_q^*$ deve essere scelto uniformemente a caso. Se una sorgente povera di randomness porta ad un K predicibile, Adv può calcolare da (r, s) la chiave privata x!

Infatti,

$$\overset{\text{noto}}{s} = \overset{\text{noto}}{k}^{-1} (\overset{\text{noto}}{H(m)} + \underline{\underline{x}} \cdot \overset{\text{noto}}{r}) \bmod q$$

Pertanto, se k è noto, l'unica incognita è x.

Anche se k è impredicibile ma usato per produrre due firme, la chiave privata x può essere recuperata!

(r, s_1) ed (r, s_2) firme su m_1 ed m_2

Possiamo procedere come segue:

$$s_1 = k^{-1}\,(H(m_1) + x\,2)\ \mathrm{mod}\ q$$

$$s_2 = k^{-1}\,(H(m_2) + x\,2)\ \mathrm{mod}\ q$$

$$\Rightarrow \quad s_1 - s_2 = k^{-1}\,(H(m_1) - H(m_2))\ \mathrm{mod}\ q$$

$\Rightarrow$ k può essere calcolato

$\Rightarrow$ x può essere calcolata procedendo come nel caso precedente.

(Applicato per ottenere la master Key della Sony Playstation PS3 nel 2010)

Firme digitali tramite funzioni hash

Può sembrare sorprendente ma schemi di firme digitali possono essere ottenuti usando funzioni hash crittografiche, senza la necessità di assunzioni di teoria dei numeri.

Schema di Lamport: sicuro per un uso singolo (one-time signature)

Signing $m = 011$:

$$sk = \begin{pmatrix} \boxed{x_{1,0}} & x_{2,0} & x_{3,0} \\ x_{1,1} & \boxed{x_{2,1}} & \boxed{x_{3,1}} \end{pmatrix} \Rightarrow \sigma = (x_{1,0},\ x_{2,1},\ x_{3,1})$$

Verifying for $m = 011$ and $\sigma = (x_1, x_2, x_3)$:

$$\left. pk = \begin{pmatrix} \boxed{y_{1,0}} & y_{2,0} & y_{3,0} \\ y_{1,1} & \boxed{y_{2,1}} & \boxed{y_{3,1}} \end{pmatrix} \right\} \Rightarrow \begin{array}{l} H(x_1) \stackrel{?}{=} y_{1,0} \\ H(x_2) \stackrel{?}{=} y_{2,1} \\ H(x_3) \stackrel{?}{=} y_{3,1} \end{array}$$

FIGURE 12.3: The Lamport scheme used to sign the message $m = 011$.

One-time signature

The one-time signature experiment $\mathsf{Sig\text{-}forge}^{\mathsf{1\text{-}time}}_{\mathcal{A},\Pi}(n)$**:**

1. $\mathsf{Gen}(1^n)$ *is run to obtain keys* (pk, sk).
2. *Adversary* $\mathcal{A}$ *is given* pk *and asks a* single *query* m' *to its oracle* $\mathsf{Sign}_{sk}(\cdot)$. $\mathcal{A}$ *then outputs* (m, σ) *with* $m \neq m'$.
3. *The output of the experiment is defined to be* 1 *if and only if* $\mathsf{Vrfy}_{pk}(m, \sigma) = 1$.

DEFINITION 12.14 *Signature scheme* $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ *is* existentially unforgeable under a single-message attack, *or is a* one-time-secure signature scheme, *if for all probabilistic polynomial-time adversaries* $\mathcal{A}$, *there exists a negligible function* negl *such that:*

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

CONSTRUCTION 12.15

Let $H : \{0,1\}^* \to \{0,1\}^*$ be a function. Construct a signature scheme for messages of length $\ell = \ell(n)$ as follows:

- Gen: on input 1^n, proceed as follows for $i \in \{1, \ldots, \ell\}$:

  1. Choose uniform $x_{i,0}, x_{i,1} \in \{0,1\}^n$.
  2. Compute $y_{i,0} := H(x_{i,0})$ and $y_{i,1} := H(x_{i,1})$.

  The public key pk and the private key sk are

$$pk = \begin{pmatrix} y_{1,0} & y_{2,0} & \cdots & y_{\ell,0} \\ y_{1,1} & y_{2,1} & \cdots & y_{\ell,1} \end{pmatrix} \quad sk = \begin{pmatrix} x_{1,0} & x_{2,0} & \cdots & x_{\ell,0} \\ x_{1,1} & x_{2,1} & \cdots & x_{\ell,1} \end{pmatrix}.$$

- Sign: on input a private key sk as above and a message $m \in \{0,1\}^\ell$ with $m = m_1 \cdots m_\ell$, output the signature $(x_{1,m_1}, \ldots, x_{\ell,m_\ell})$.
- Vrfy: on input a public key pk as above, a message $m \in \{0,1\}^\ell$ with $m = m_1 \cdots m_\ell$, and a signature $\sigma = (x_1, \ldots, x_\ell)$, output 1 if and only if $H(x_i) = y_{i,m_i}$ for all $1 \leq i \leq \ell$.

The Lamport signature scheme.

Teorema. Se H è one-way, la costruzione è sicura.