



Digital Signatures

Murat Kantarcioglu



Digital Signatures

★ **Define** a digital signature scheme $DS = (\mathcal{K}, \text{Sign}, VF)$

★ **Key generation:** $(pk, sk) \xleftarrow{\$} \mathcal{K}$

★ **Signing a message:** $\sigma \xleftarrow{\$} \text{Sign}_{sk}(M)$

★ **Signature Verification** $d \xleftarrow{\$} VF_{pk}(M, \sigma)$

★ $d = 1$ if σ is valid for
for given message under (pk, sk) pair

★ else $d = 0$



Digital Signature Assumptions

Alice generates (pk, sk)



$(M, \sigma \leftarrow \text{Sig}_{sk}(M))$



Bob has **correct** pk

Bob **outputs** $VF_{pk}(M, \sigma)$

- ★ Bob assumed to have **correct** pk
- ★ Sender (Alice) has the **private key**
- ★ Sig could be randomized and /or stateful
- ★ We will mainly focus on deterministic Sig algorithms
 - ▶ Unlike PKE algorithms



Defining Security

Definition 9.2 Let $\mathcal{DS} = (\mathcal{K}, \text{Sign}, \text{VF})$ be a digital signature scheme, and let A be an algorithm that has access to an oracle and returns a pair of strings. We consider the following experiment:

Experiment $\text{Exp}_{\mathcal{DS}}^{\text{uf-cma}}(A)$

$(pk, sk) \xleftarrow{\$} \mathcal{K}$

$(M, \sigma) \leftarrow A^{\text{Sign}_{sk}(\cdot)}(pk)$

If the following are true return 1 else return 0:

- $\text{VF}_{pk}(M, \sigma) = 1$
- $M \in \text{Messages}(pk)$
- M was not a query of A to its oracle

The *uf-cma-advantage* of A is defined as

$$\text{Adv}_{\mathcal{DS}}^{\text{uf-cma}}(A) = \Pr \left[\text{Exp}_{\mathcal{DS}}^{\text{uf-cma}}(A) = 1 \right]. \blacksquare$$

pk (M, σ) for any M

$A:$

M_1, σ_1

M_2, σ_2

\dots

M_q, σ_q

$M \in \{M_1, \dots, M_q\}$



RSA based Signatures

★ $((N, e), (N, p, q, d)) \leftarrow (K)$ where $e.d = 1 \pmod{\phi(N)}$, $N = pq$

★ Signature Generation

- ▶ Algorithm $Sign_{N,p,q,d}(M)$
- ▶ if $M \in Z_N^*$ return \perp
- ▶ return $M^d \pmod N$

★ Verification

- ▶ Algorithm $VF_{N,e}(M, \sigma)$
- ▶ if $M \notin Z_N^* \vee \sigma \notin Z_N^*$ return 0
- ▶ if $M = \sigma^e \pmod N$ return 1 else 0

★ Direct RSA signature generation is **not secure**

Possible Attacks

★ Forger F_1

- ▶ Forger $F_1^{Sign_{N,p,q,d}(\cdot)}(N, e)$
- ▶ return $(1, 1)$

$1^d \bmod N = 1$
 $(1, 1)$

$Adv_{OS}^{uf-cma}(F_1) = 1$

All attacks have **advantage one**

★ Forger F_2

- ▶ Forger $F_2^{Sign_{N,p,q,d}(\cdot)}(N, e)$
- ▶ $\sigma \leftarrow Z_N^*$, $M \leftarrow \sigma^e \bmod N$
- ▶ return (M, σ)

$m^d, \sigma \Rightarrow (\sigma^e)^d = \sigma$

$Adv_{OS}^{uf-cma}(F_2) = 1$

★ Forger F_3

- ▶ Forger $F_3^{Sign_{N,p,q,d}(\cdot)}(N, e)$
- ▶ $M_1 \leftarrow Z_N^* - \{1, M\}$, $M_2 \leftarrow MM_1^{-1} \bmod N$
- ▶ $\sigma_1 \leftarrow Sign_{N,p,q,d}(M_1)$, $\sigma_2 \leftarrow Sign_{N,p,q,d}(M_2)$
- ▶ return $(M, \sigma_1 \sigma_2 \bmod N)$



Hash-then-invert paradigm

- ★ **Goal:** RSA based scheme that
 - ▶ is **provably secure**
 - ▶ has **Flexible** message space

- ★ **Idea** Hash the message first given $H_N : \{0, 1\}^* \mapsto Z_N^*$

- ★ **Signature Generation**
 - ▶ Algorithm $Sign_{N,p,q,d}(M)$
 - ▶ $y \leftarrow H_N(M)$
 - ▶ return $y^d \bmod N$

- ★ **Verification**
 - ▶ Algorithm $VF_{N,e}(M, \sigma)$
 - ▶ $y \leftarrow H_N(M)$
 - ▶ if $y = \sigma^e \bmod N$ return 1 else 0



Hash then Invert Paradigm

- ★ Previous Forgers described do **not work well** for Hash-then-Invert
 - ▶ $H_N(1) \neq 1$ with high probability (w.h.p)
 - ▶ $\sigma^e \bmod N \neq H_N(M)$ w.h.p
 - ▶ $H_N(M_1).H_N(M_2) \neq H_N(M)$ w.h.p
- ★ **Not secure** if it is easy to find $M_1 \neq M_2$ such that $H_N(M_1) = H_N(M_2)$
- ★ What are the **assumptions** needed to make Hash then Invert Paradigm Secure??



Full Domain Hash RSA signatures

★ $H : \{0, 1\}^* \mapsto Z_N^*$ is a **random function** known by everybody

★ Signature Generation

- ▶ Algorithm $Sign_{N,p,q,d}^{H(\cdot)}(M)$
- ▶ $y \leftarrow H(M)$
- ▶ return $y^d \bmod N$

★ Verification

- ▶ Algorithm $VF_{N,e}^{H(\cdot)}(M, \sigma)$
- ▶ $y \leftarrow H_N(M)$
- ▶ if $y = \sigma^e \bmod N$ return 1 else 0

Experiment $\text{Exp}_{DS}^{\text{uf-cma}}(F)$
 $((N, e), (N, p, q, d)) \xleftarrow{\$} \mathcal{K}_{\text{rsa}}$
 $[H \xleftarrow{\$} \text{Func}(\{0, 1\}^*, Z_N^*)$
 $(M, x) \xleftarrow{\$} F^{H(\cdot), \text{Sign}_{N,p,q,d}^{H(\cdot)}}(N, e)$
If the following are true return 1 else return 0:
- $\text{VF}_{pk}^H(M, \sigma) = 1$
- M was not a query of A to its oracle



FDH-RSA

- ★ Consider **adversaries** running in time t , making q_{sig} oracle queries and at most q_h hash queries
- ★ **Simulate** the random H by choosing random answers and storing them on a table
 - ▶ Function $H(x)$
 - ▶ If $T(x) \neq \text{Null}$ Then $T(x) \stackrel{\$}{\leftarrow} Z_N^*$
 - ▶ Return $T[x]$
- ★ **Thm:** Let FDH-RSA in the random oracle model described as before. Let F be an adversary attacking FDH-RSA making q_{sig} signature queries, q_h hash queries. Then \exists an Adversary I

$$\left(Adv_{DS}^{uf-cma}(F) \leq q_h \cdot Adv_{K_{rsa}}^{ow-kea}(I) \right)$$



Pr of d Thm

★ **Note** I is given $(N, e), y$ and tries to find x s.t.
 $x^e \bmod N$ $y = x^e \bmod N$

★ I will **run** F to find the x

★ I will **answer** F 's oracle queries to H and $Sign$ as it wishes

★ I will **use** the F to invert y

★ **Idea:** I modifies answers to F 's oracle queries to invert y



Pr of d Thm

Inverter $I(N, e, y)$

Initialize arrays $Msg[1 \dots q_{hash}]$, $X[1 \dots q_{hash}]$, $Y[1 \dots q_{hash}]$ to empty

$j \leftarrow 0$; $i \xleftarrow{s} \{1, \dots, q_{hash}\}$

Run F on input (N, e)

If F makes oracle query $(hash, M)$

then $h \leftarrow H-Sim(M)$; return h to F as the answer

If F makes oracle query $(sign, M)$

then $x \leftarrow Sign-Sim(M)$; return x to F as the answer

Until F halts with output (M, x)

$y' \leftarrow H-Sim(M)$

Return x

y'

$H-Sim(M)$

$H(M), x^d \pmod N$

$Msg[j]$ – The j -th hash query in the experiment

$Y[j]$ – The reply of the hash oracle simulator to the above, meaning the value playing the role of $H(Msg[j])$. For $j = i$ it is y .

$X[j]$ – For $j \neq i$, the response to sign query $Msg[j]$, meaning it satisfies $(X[j])^e \equiv Y[j] \pmod N$. For $j = i$ it is undefined.



Pr of d Thm

M

Subroutine *H-Sim*(*v*)

$l \leftarrow \text{Find}(\text{Msg}, v, j)$; $j \leftarrow j + 1$; $\text{Msg}[j] \leftarrow v$

If $l = 0$ then

If $j = i$ then $Y[j] \leftarrow y$

Else $X[j] \leftarrow Z_N^s$; $Y[j] \leftarrow (X[j])^e \text{ mod } N$

EndIf

Return $Y[j]$

Else

If $j = i$ then abort

Else $X[j] \leftarrow X[l]$; $Y[j] \leftarrow Y[l]$; Return $Y[j]$

EndIf

EndIf

★ *Find*(*A*, *v*, *j*)

▶ if $\exists l \leq j, A[l] = v$ return 0

▶ else smallest l where $A[l] = v$

$A = [1, 3, 3, 5, 7]$

$\text{Find}(A, 3, 4) = 0$

Subroutine *Sign-Sim*(*M*)

$h \leftarrow H\text{-Sim}(M)$

If $j = i$ then abort

Else return $X[j]$

EndIf

$X[5] \leftarrow r$ $Y[5] \leftarrow r^e \text{ mod } N$

Find(*A*, *v*, *j*)

for ($i = 1$ to \bar{j})

if ($A[i] == v$)

return i ;

return 0.

$\text{Find}(A, 3, 3) = 2$



Pr of d Thm.

★ Inside $H - sim(v)$, if $l = 0$ and $j \neq i$ $X[j] \leftarrow Z_N^*$
and $Y[j] \leftarrow (X[j])^e \pmod N$ and returns $Y[j]$

★ Sign-sim(M) returns $X[j]$

$$y \leftarrow H(M) \xrightarrow{d} y = x^d \pmod n$$

$$\begin{aligned} Pr[I \text{ inverts } y] &= Pr[I \text{ inverts } y | \text{no abort}] \cdot Pr[\text{no abort}] \\ &\quad + Pr[I \text{ inverts } y | \text{abort}] \cdot Pr[\text{abort}] \\ &= Pr[I \text{ inverts } y | \text{no abort}] \cdot Pr[\text{no abort}] \\ &\geq Adv_{DS}^{uf-cma}(F) \cdot \frac{1}{q_{hash}} \end{aligned}$$

when \underline{I} calls the $H-sim$ last time
if $\text{find}(Msg, M, q_{hash}) = l$ & $l = i$

★ $H : \{0, 1\}^* \mapsto Z_N^*$ is a **random function** known by everybody

★ Signature Generation

- ▶ Algorithm $Sign_{N,p,q,d}^{H(\cdot)}(M)$
- ▶ $r \xleftarrow{\$} \{0, 1\}^s$
- ▶ $y \leftarrow H(r||M)$
- ▶ return $(r, y^d \bmod N)$

$H(M)$

$H(r||M)$

★ Verification

- ▶ Algorithm $VF_{N,e}^{H(\cdot)}(M, \sigma)$
- ▶ Parse σ as (r, x)
- ▶ $y \leftarrow H(r||M)$
- ▶ if $y = x^e \bmod N$ return 1 else 0

Theorem 9.4 Let \mathcal{DS} be the PSS0 scheme with security parameters k and s . Let F be an adversary making q_{sig} signing queries and $q_{\text{hash}} \geq 1 + q_{\text{sig}}$ hash oracle queries. Then there exists an adversary I such that

$$\text{Adv}_{\mathcal{DS}}^{\text{uf-cma}}(F) \leq \text{Adv}_{\mathcal{K}_{\text{rsa}}}^{\text{ow-kea}}(I) + \frac{(q_{\text{hash}} - 1) \cdot q_{\text{sig}}}{2^s} . \blacksquare \quad (9.3)$$



El-Gamal Signature Scheme

★ Define a digital signature scheme $DS = (\mathcal{K}, Sign, VF)$

★ Key generation: $((p, \alpha, y), (p, a)) \xleftarrow{\$} \mathcal{K}$ Where $\alpha^a = \underline{(y) \bmod p}$ and α is a generator of Z_p^*

★ Signing a message M

- ▶ Select $k \in Z_p^*$ with $\gcd(k, p-1) = 1$
- ▶ $r \leftarrow \alpha^k, s \leftarrow k^{-1}(H(M) - ar) \bmod (p-1)$
- ▶ return (r, s)

$H(M || r)$

★ Signature Verification for $(M, (r, s))$

- ▶ $v_1 \leftarrow y^r r^s \bmod p$
- ▶ $v_2 \leftarrow \alpha^{H(m)} \bmod p$
- ▶ Accept if $v_1 = v_2$

$$y^r = \alpha^{ar} \bmod p$$

$$r^s = \alpha^{k^{-1}(H(m) - ar)}$$

$$y^r \cdot r^s = \alpha^{ar + k^{-1}(H(m) - ar)} = \alpha^{H(m)} \bmod p$$



The Digital Signature Algorithm (DSA)

★ Key Generation:

- ▶ Select a prime $2^{159} < q < 2^{160}$
- ▶ Choose $t \leq 8$ and a prime p where $2^{511+64t} < p < 2^{512+64t}$ and $q|(p-1)$
- ▶ Select a random $b \in \mathbb{Z}_p^*$ s.t. $\alpha \leftarrow b^{(p-1)/q} \bmod p$ and $\alpha \neq 1 \bmod p$
- ▶ Select a random integer a s.t. $1 \leq a \leq q-1$
- ▶ Compute $y \leftarrow \alpha^a \bmod p$
- ▶ Public key is (p, q, α, y) , private key is \underline{a}

★ Signature Generation: for message M

- ▶ Select a random k s.t. $0 < k < q$
- ▶ $r \leftarrow (\alpha^k \bmod p) \bmod q$
- ▶ $s \leftarrow k^{-1}(H(M) + ar) \bmod q$
- ▶ return (r, s)

$$s \leftarrow k^{-1}(H(M) + ar)$$

★ Verification for $(M, (r, s))$

- ▶ Check that $0 < r < q$ and $0 < s < q$
- ▶ $u_1 \leftarrow s^{-1} \cdot H(M)$ and $u_2 = r s^{-1} \pmod q$
- ▶ $v \leftarrow (\alpha^{u_1} \cdot y^{u_2} \pmod p) \pmod q$
- ▶ Accept iff $v = r$

$$s = k^{-1} (H(m) + ar)$$

$$r = (\alpha^k \pmod p) \pmod q$$

$$\begin{aligned}
 \checkmark \in \alpha & \quad s^{-1} \cdot H(m) \quad a r s^{-1} & = & \quad s^{-1} (H(m) + ar) \\
 & \quad \cdot \alpha & = & \quad \alpha \\
 & & = & \quad \left(k^{-1} \cdot (H(m) + ar) \right)^{-1} \cdot (H(m) + ar) \\
 & & = & \quad \alpha^k (H(m) + ar)^{-1} \cdot (H(m) + ar) \\
 & & = & \quad \alpha^k \\
 & & = & \quad \alpha^k
 \end{aligned}$$



Schnorr Scheme

- ★ **Key generation** is the same as DSA except no restriction on (p, q)
- ★ **Signature generation** for M
 - ▶ Choose random secret k , $1 \leq k \leq q - 1$
 - ▶ $r \leftarrow \alpha^k \bmod p$, $e \leftarrow H(M||r)$, $s \leftarrow ae + k \bmod q$
 - ▶ return (s, e)
- ★ **Signature verification** for $(M, (s, e))$
 - ▶ $v \leftarrow \alpha^s y^{-e} \bmod p$ and $e = H(M||v)$
 - ▶ Accept iff $v = e$