Modern Public Key Cryptography

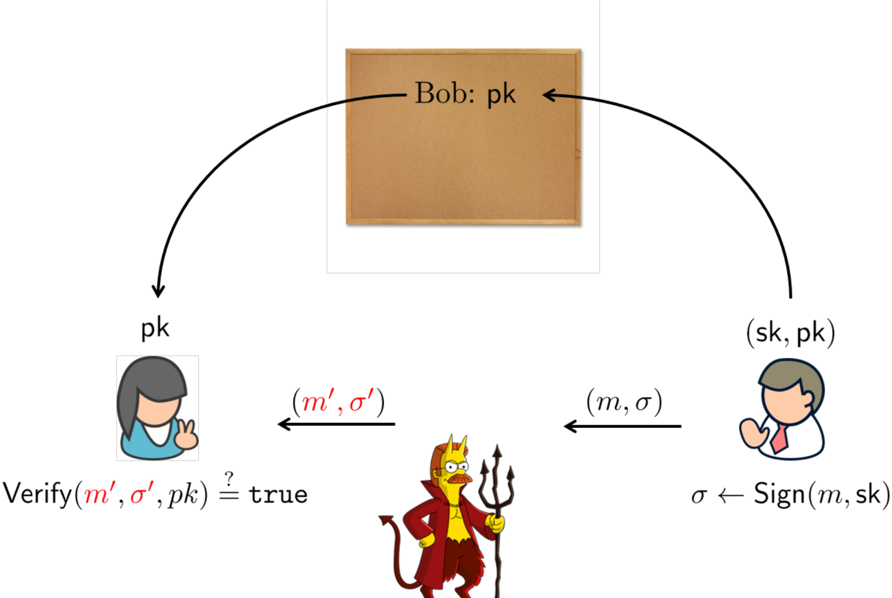
Digital Signature Schemes

Daniel Kales based on slides by Sebastian Ramacher and David Derler

March 23, 2022

SCIENCE
PASSION
TECHNOLOGY

Digital Signatures - The Setting



Formal Definition

Signature Scheme

KeyGen(1^κ): Given security parameter κ, outputs a key pair (sk, pk) (*pk* fixes M_κ)

Sign(m, sk): Given msg $m \in M_\kappa$ and signing key sk, computes signature σ on m using sk and outputs σ

Verify(m, σ, pk): Given msg $m \in M_\kappa$, σ and public key pk, returns 1 if (m, σ) is a valid msg-sig pair under pk and 0 otherwise

Algorithm Sign may also be stateful (not considered here)

Security

Correctness

$$\forall \kappa, (\mathsf{sk}, \mathsf{pk}) \leftarrow KeyGen(1^\kappa), m \in M_\kappa :$$
$$\Pr[\mathsf{Verify}(m, \mathsf{Sign}(m, \mathsf{sk}), \mathsf{pk})] = 1 - \epsilon(\kappa)$$

How to define when a scheme is secure?

- An adversary should not able to forge valid message/signature pairs
- Even when interacting with an honest signer in some way
  - What does forge and interacting mean?
  - We do not incorporate any semantics (e.g., what is a meaningful message?)

Overview of Target and Attacks

Targets (hardest to easiest)

- Total break: Obtain the secret signing key
- Selective forgery: Produce signature for some selected message(s)
- (Weak) Existential forgery: Produce at least one valid signature for a message where no signature was previously requested
- Strong existential forgery: Produce a valid signature different from any previously seen signature

Overview of Target and Attacks

Attacks (weak to strong)

- No-message attack: Only access to the public key
- Random-message attack: Obtain signatures for random message (no control over messages)
- Known-message attack: Access to a list of signatures (messages chosen before seeing public key)
- Chosen-message attack: Access to a list of signatures (messages chosen after seeing the public key)
- Adaptively chosen-message attack: Obtain signatures for any message

Overview of Target and Attacks

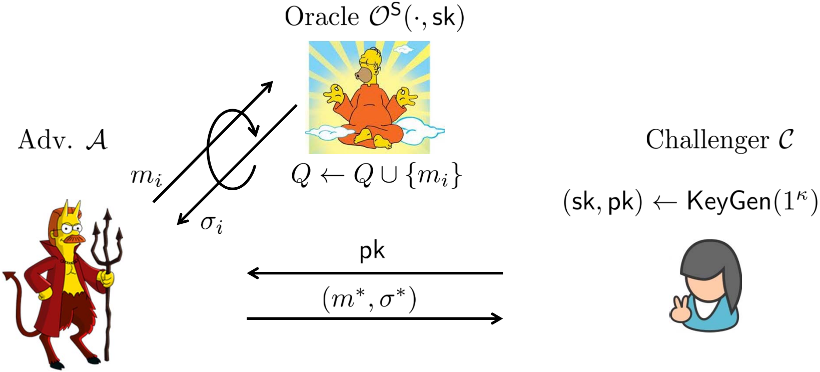
- Another dimension is the number of signatures accessible to an adversary
  - A single signature (one-time)
  - Unbounded number of signatures
- Highest security guarantees if strongest attacker can not even achieve easiest target
  - Existential unforgeability under adaptively chosen message attacks (EUF-CMA)
  - Usually weak existential unforgeability

Overview of Target and Attacks

- Another dimension is the number of signatures accessible to an adversary
  - A single signature (one-time)
  - Unbounded number of signatures
- Highest security guarantees if strongest attacker can not even achieve easiest target
  - Existential unforgeability under adaptively chosen message attacks (EUF-CMA)
  - Usually weak existential unforgeability

EUF-CMA

Experiment $\mathbf{Exp}^{\text{EUF-CMA}}_{\Sigma,\mathcal{A}}(\cdot)$:



if $m^* \notin Q \wedge \mathsf{Verify}(m^*, \sigma^*, \mathsf{pk}) = \mathtt{true}$ return 1;
else return 0;

EUF-CMA

Definition (Existential Unforgeability Under Chosen Message Attacks (EUF-CMA))

The advantage $\mathsf{Adv}^{\mathcal{A}}_{\mathsf{EUF\text{-}CMA}}(\cdot)$ of an adversary $\mathcal{A}$ in the EUF-CMA experiment as

$$\mathsf{Adv}^{\mathcal{A}}_{\mathsf{EUF\text{-}CMA}}(\kappa) = \Pr\left[\begin{array}{l}(\mathsf{sk},\mathsf{pk}) \leftarrow \mathsf{KeyGen}(1^{\kappa}),\\ (m^*,\sigma^*) \leftarrow \mathcal{A}^{\mathsf{Sig}(\mathsf{sk},\cdot)}(\mathsf{pk})\end{array} : \begin{array}{r} m^* \notin \mathcal{Q}^{\mathsf{Sig}} \wedge \\ \mathsf{Verify}(\mathsf{pk},m^*,\sigma^*) = 1\end{array}\right],$$

where the environment maintains an initially empty list $\mathcal{Q}^{\mathsf{Sig}}$ and the oracles are defined as follows:

$\mathsf{Sig}(\mathsf{sk},m)$: Set $\mathcal{Q}^{\mathsf{Sig}} \leftarrow \mathcal{Q}^{\mathsf{Sig}} \cup \{m\}$ and return $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk},m)$.

A signature scheme is secure against EUF-CMA attacks, if for every PPT adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathcal{A}}_{\mathsf{EUF\text{-}CMA}}(\cdot)$ is negligible.

What About Textbook RSA Signatures?

- Plain RSA: $pk = (N, e)$ and $sk = (N, d)$
  - To sign $m \in \mathbb{Z}_N$ compute $\sigma \leftarrow m^d \bmod N$
  - To verify given (m, σ) check if $\sigma^e \equiv m \pmod{N}$
- Choose $\sigma \xleftarrow{R} \mathbb{Z}_N$ and set $m \leftarrow \sigma^e \bmod N$
  - The pair (m, σ) is a valid signature!
  - Existential forgery under no-message attack
  - Also other attacks (homomorphism)
- Use of RSA-FDH/RSA-PSS

What About Textbook RSA Signatures?

- Plain RSA: $pk = (N, e)$ and $sk = (N, d)$
  - To sign $m \in \mathbb{Z}_N$ compute $\sigma \leftarrow m^d \bmod N$
  - To verify given (m, σ) check if $\sigma^e \equiv m \pmod{N}$
- Choose $\sigma \xleftarrow{R} \mathbb{Z}_N$ and set $m \leftarrow \sigma^e \bmod N$
  - The pair (m, σ) is a valid signature!
  - Existential forgery under no-message attack
  - Also other attacks (homomorphism)
- Use of RSA-FDH/RSA-PSS

RSA-Full-Domain Hash (RSA-FDH)

Scheme

KeyGen(1^κ): Output public and private RSA keys $(\mathsf{pk}, \mathsf{sk}) \leftarrow ((N, e), d)$. Specify function $H : \{0,1\}^* \rightarrow \mathbb{Z}_N^*$.

Sign(m, sk): Return signature $\sigma \leftarrow (H(m))^d \mod N$

Verify(m, σ, pk): Return $[\sigma^e == H(m)]$

RSA-Full-Domain Hash (RSA-FDH)

Scheme

KeyGen(1^κ): Output public and private RSA keys $(\mathrm{pk}, \mathrm{sk}) \leftarrow ((N, e), d)$. Specify function $H : \{0,1\}^* \rightarrow \mathbb{Z}_N^*$.

Sign(m, sk): Return signature $\sigma \leftarrow (H(m))^d \mod N$

Verify(m, σ, pk): Return $[\sigma^e == H(m)]$

RSA-Full-Domain Hash (RSA-FDH)

Scheme

KeyGen(1^κ): Output public and private RSA keys $(\mathrm{pk}, \mathrm{sk}) \leftarrow ((N, e), d)$. Specify function $H : \{0,1\}^* \rightarrow \mathbb{Z}_N^*$.

Sign(m, sk): Return signature $\sigma \leftarrow (H(m))^d \mod N$

Verify(m, σ, pk): Return $[\sigma^e == H(m)]$

How to Prove RSA-FDH is EUF-CMA secure in the ROM?

Outline

- Suppose $\mathcal{A}$ breaks EUF-CMA security of RSA-FDH with non-negligible probability
- Then, we try to build adversary $\mathcal{A}'$ breaking the RSA assumption, i.e.,

$$\text{given } (N, e, c) \text{ try to find } c^d = m \mod N.$$

How to Prove This? (ctd)

Recap: RSA Assumption (OW-RSA)

Given (N, e), it is hard to invert $f(x) = x^e \mod N$

- i.e. given $y \in \mathbb{Z}_N^*$ hard to find $x \in \mathbb{Z}_N^*$ s.t. $y = x^e \mod N$
- (given that (N, e) fulfills usual criteria in this context (cf. last VO))

Proof: RSA-FDH

Proof Sketch (Coron, 2000 [4])

$\mathcal{A}'$ gets input (N, e, c), starts $\mathcal{A}$ on $\mathrm{pk} \leftarrow (N, e)$ and simulates RO and EUF-CMA environment for $\mathcal{A}$:

- When $\mathcal{A}$ queries RO for m, $\mathcal{A}'$ picks $r \xleftarrow{R} \mathbb{Z}_N^*$, computes hash $h \leftarrow r^e \mod N$ with probability p and $h \leftarrow c \cdot r^e \mod N$ with probability $1 - p$, stores (m, h, r) and returns h
- When $\mathcal{A}$ queries signature for m, $\mathbb{R}$ gets (m, h, r) and returns r if $h = r^e \mod N$ and aborts otherwise
- If $\mathcal{A}$ returns forgery (m^*, σ^*) [1] s.t. $H(m^*) = h^* = c \cdot (r^*)^e \mod N, \sigma^* = c^d \cdot r^* \mod N$. $\mathcal{A}'$ returns $c^d = \sigma^* / r^* \mod N$

[1]Observe: to compute σ^*, $\mathcal{A}$ must have queried RO on m^*

Proof: RSA-FDH (ctd)

Analysis

- Values h look random to $\mathcal{A}$, making simulation of RO perfect, as
  - values r random
  - $\mathcal{A}$ has never seen c directly
- Simulation of signatures perfect (according to previous observation)

Proof: RSA-FDH (ctd)

Analysis

- Values h look random to $\mathcal{A}$, making simulation of RO perfect, as
  - values r random
  - $\mathcal{A}$ has never seen c directly
- Simulation of signatures perfect (according to previous observation)

Proof: RSA-FDH (ctd)

Analysis (ctd)

- Simulation works with prob. p^q (for q signature queries)
- If simulation ok, $\mathcal{A}'$ can use forgery with prob. $1 - p$
- If $\mathcal{A}$ succeeds with non-negligible prob. $\epsilon(\kappa)$, $\mathbb{R}$ succeeds with non-negligible prob. $(1 - p)p^q\epsilon(\kappa)$ and asymptotically: $O(\frac{\epsilon(\kappa)}{q})$
- Reduction not always successful: Security loss by q

Proof: RSA-FDH (ctd)

Analysis (ctd)

- Simulation works with prob. p^q (for q signature queries)
- If simulation ok, $\mathcal{A}'$ can use forgery with prob. $1 - p$
- If $\mathcal{A}$ succeeds with non-negligible prob. $\epsilon(\kappa)$, $\mathbb{R}$ succeeds with non-negligible prob. $(1 - p)p^q\epsilon(\kappa)$ and asymptotically: $O(\frac{\epsilon(\kappa)}{q})$
- Reduction not always successful: Security loss by q

Proof: RSA-FDH (ctd)

Analysis (ctd)

- Simulation works with prob. p^q (for q signature queries)
- If simulation ok, $\mathcal{A}'$ can use forgery with prob. $1 - p$
- If $\mathcal{A}$ succeeds with non-negligible prob. $\epsilon(\kappa)$, $\mathbb{R}$ succeeds with non-negligible prob. $(1 - p)p^q\epsilon(\kappa)$ and asymptotically: $O(\frac{\epsilon(\kappa)}{q})$
- Reduction not always successful: Security loss by q

Proof: RSA-FDH (ctd)

Analysis (ctd)

- Simulation works with prob. p^q (for q signature queries)
- If simulation ok, $\mathcal{A}'$ can use forgery with prob. $1 - p$
- If $\mathcal{A}$ succeeds with non-negligible prob. $\epsilon(\kappa)$, $\mathbb{R}$ succeeds with non-negligible prob. $(1 - p)p^q\epsilon(\kappa)$ and asymptotically: $O(\frac{\epsilon(\kappa)}{q})$
- Reduction not always successful: Security loss by q

Message Length Extension

- We have associated a message space M_κ related to the security parameter κ to any scheme Σ
- How can we extend the message space to (nearly) arbitrary message sizes?
  - Block-wise signing (not efficient)
  - Hash-then-sign paradigm (very efficient)

Hash-Then-Sign Paradigm

- Let Σ' be: Use hash function H to map any arbitrary length message m to M_κ before applying Sign of Σ

Theorem

If Σ is EUF-CMA secure and H is collision resistant, then Σ' is EUF-CMA secure

Proof Sketch.

Let $m_1, \ldots, m_\ell$ be the messages queried by $\mathcal{A}$ and (m^*, σ^*) the valid forgery

Case 1. $H(m^*) = H(m_i)$ for some $i \in [\ell]$: we have a collision for H

Case 2. $H(m^*) \neq H(m_i)$ for all $i \in [\ell]$: we have that $(H(m^*), \sigma^*)$ is a forgery for Σ

□

Constructions

- Constructions based on general assumption (not covered)
  - OWFs imply sEUF-CMA secure schemes
  - "Hash-based" signatures (post-quantum)
- Constructions in the ROM
  - Have already seen RSA-FDH
  - Will look at pairing-based version
- Constructions in the SM
  - see "Further Reading"

Generic Compilers for Strong Security

- CMA from RMA
  - Split m into m_L and m_R for $m_L \xleftarrow{R} \{0,1\}^k$ such that $m = m_L \oplus m_R$
  - Sign $r||m_L$ and $r||m_R$ with two independent keys of Σ, where $r \xleftarrow{R} \{0,1\}^k$
- CMA from KMA
  - Let Σ be a KMA-secure scheme, Σ' be a KMA-secure one-time scheme. Generate a long-term key-pair for Σ
  - For message m generate one-time key of Σ' and sign m with one-time key. Sign one-time public key using long-term signing key
- CMA from IBE
- CMA in RO from ID schemes (Fiat-Shamir)

BLS Signatures

"Bilinear" analogue to RSA-FDH scheme. Let $H : \{0,1\}^k \to \mathbb{G}$.

Scheme

KeyGen(1^κ): Choose $\mathcal{G}^\kappa$ and $x \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$ and set sk $\leftarrow x$ and pk $\leftarrow y = g^x$

Sign(m, sk): Compute $h = H(m)$ and output $\sigma \leftarrow h^x$

Verify(m, σ, pk): Return $[e(\sigma, g) = e(H(m), y)]$

Very short signatures. Signature valid if $(H(m), y, \sigma)$ is DDH tuple

BLS Signatures

Theorem

If CDH assumption holds in $\mathbb{G}$ and H is a random oracle, then BLS is sEUF-CMA secure.

- Proof nearly identical to RSA-FDH proof
- For non-tight reduction
  - Obtain CDH instance (h, y)
  - Guess index $i \in [q_H]$ of RO query
  - Embed h into i^{th} query and hope forgery (m^*, σ^*) is for m_i
  - If $m^* = m_i$ output σ^* as CDH solution
- Works also with Coron's strategy (tighter reduction; see RSA-FDH proof)

What you should know...

- Security models for digital signature schemes
  - Types of forgeries and attacks
- RSA-FDH proof idea
- Message length extension (hash-then-sign)
- Generic compilers from RMA/KMA

Questions?

Further Reading I

[1] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway.

Relations Among Notions of Security for Public-Key Encryption Schemes.

In *Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings*, pages 26–45, 1998.

[2] Dan Boneh and Xavier Boyen.

Short Signatures Without Random Oracles and the SDH Assumption in Bilinear Groups.

J. Cryptology, 21(2):149–177, 2008.

[3] Dan Boneh, Ben Lynn, and Hovav Shacham.

Short Signatures from the Weil Pairing.

J. Cryptology, 17(4):297–319, 2004.

Further Reading II

[4] Jean-Sébastien Coron.

On the exact security of full domain hash.

In Mihir Bellare, editor, *Advances in Cryptology - CRYPTO 2000, 20th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2000, Proceedings*, volume 1880 of *Lecture Notes in Computer Science*, pages 229–235. Springer, 2000.

[5] Rosario Gennaro, Shai Halevi, and Tal Rabin.

Secure hash-and-sign signatures without the random oracle.

In *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, pages 123–139, 1999.

[6] Susan Hohenberger and Brent Waters.

Short and stateless signatures from the RSA assumption.

In *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, pages 654–670, 2009.

Further Reading III

[7] Jonathan Katz.

Digital Signatures.

Springer, 2010.

[8] Brent Waters.

Efficient identity-based encryption without random oracles.

In *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, pages 114–127, 2005.