Introduction to Cryptology

12.1 - Schnorr and DSA/ECDSA

Federico Pintore

Mathematical Institute, University of Oxford (UK)

UNIVERSITY OF OXFORD

Michaelmas term 2020

Identification Protocols

Interactive protocols used for authentication.

The party who identify themselves is called *prover*; the party that verifies the identity is called *verifier*.

The prover has a public key and its corresponding secret key. The verifier only knows the prover's public key.

We consider three-round identifications protocols:

- the prover is specified by two algorithms, P_1 and P_2;
- the verifier is specified by an algorithm V.

Identification Protocols

An identification protocol $I = (\mathrm{KeyGen}, \mathrm{P}_1, \mathrm{P}_2, \mathrm{V})$ consists of four PPT algorithms that and a challenge space Ω_{ch}.

- $(\mathrm{PK}, \mathrm{SK}) \leftarrow \mathrm{KeyGen}(n)$: on input a security parameter n, it returns a public-private key pair $(\mathrm{PK}, \mathrm{SK})$.
- $(\mathrm{com}, \mathrm{st}) \leftarrow \mathrm{P}_1(\mathrm{PK})$: it takes a public key PK and returns a commitment com together with with a state st.
- $\mathrm{rsp} \leftarrow \mathrm{P}_2(\mathrm{SK}, \mathrm{st}, \mathrm{ch})$: on input a secret key SK, a state st and a challenge $\mathrm{ch} \in \Omega_{\mathrm{ch}}$, it returns a response rsp.
- $1/0 \leftarrow \mathrm{V}(\mathrm{PK}, \mathrm{com}, \mathrm{ch}, \mathrm{rsp})$: a deterministic algorithm that returns either 1 (accept) or 0 (reject).

Correctness:

$$\Pr(\mathrm{V}(\mathrm{PK}, \mathrm{com}, \mathrm{ch}, \mathrm{P}_2(\mathrm{SK}, \mathrm{st}, \mathrm{ch})) = 1) = 1.$$

Security of Identification Protocols

The Identification Experiment $\mathrm{Ident}_{\mathcal{A},I}(n)$

Security of Identification Protocols

The Identification Experiment $\mathrm{Ident}_{\mathcal{A},I}(n)$

Challenger Ch		Adversary $\mathcal{A}$
$(\mathrm{PK}, \mathrm{SK}) \leftarrow \mathrm{KeyGen}(n)$		
	$\xrightarrow{\mathrm{PK}}$	
		Access to $\mathrm{Trans}_{\mathrm{SK}}$
		$(\mathrm{com}^*, \mathrm{st}^*) := \mathrm{P}_1(\mathrm{PK})$
	$\xleftarrow{\mathrm{com}^*}$	
$\mathrm{ch}^* \leftarrow \Omega_{\mathrm{ch}}$	$\xrightarrow{\mathrm{ch}^*}$	
		Access to $\mathrm{Trans}_{\mathrm{SK}}$
		Outputs rsp^*

The oracle $\mathrm{Trans}_{\mathrm{SK}}$, when queried without any input, runs I and returns the resulting transcript $(\mathrm{com}, \mathrm{ch}, \mathrm{rsp})$.

Security of Identification Protocols

$\mathcal{A}$ wins the game, i.e. $\mathrm{Ident}_{\mathcal{A},I}(n) = 1$, if

$$\mathrm{V}(\mathrm{PK}, \mathrm{com}^*, \mathrm{ch}^*, \mathrm{rsp}^*) = 1.$$

Definition

The identification protocol I is secure against a passive attack if, for every PPT adversary $\mathcal{A}$, it holds that

$$\Pr(\mathrm{Ident}_{\mathcal{A},I}(n) = 1) \leq \mathrm{negl}(n)\,.$$

Fiat-Shamir Transform

It turns an identification protocol
into a digital signature scheme.

The signer applies a hash function H to (m, com) in order to generate the challenge ch.

The signature on m is the transcript $(\mathrm{com}, \mathrm{ch}, \mathrm{rsp})$.

The verifier checks if

- $H(m, \mathrm{com}) = \mathrm{ch}$;
- $\mathrm{V}(\mathrm{PK}, \mathrm{com}, \mathrm{ch}, \mathrm{rsp}) = 1$.

Security of the Fiat-Shamir Transform

Theorem

If I is an identification protocol secure against a passive attack and H is modelled as a random oracle, the digital signature scheme obtained by applying the Fiat-Shamir transform is existentially unforgeable.

Schnorr Identification Scheme

The Schnorr identification scheme $I_{Sch} = (\mathrm{KeyGen}, \mathrm{P}_1, \mathrm{P}_2, \mathrm{V})$ with $\Omega_{\mathrm{ch}} = \mathbb{Z}_q$ is defined as follows.

- $(\mathrm{PK}, \mathrm{SK}) \leftarrow \mathrm{KeyGen}(n)$: it runs a group generation algorithm $\mathcal{G}$ on a security parameter n, obtaining a description of a cyclic group $\mathbb{G}$ - of order q, with $||q|| = n$ - together with a generator g.

  It samples a uniform $x \in \mathbb{Z}_q$ and computes $h := g^x$. Then it sets PK to $(\mathbb{G}, q, g, h)$ and SK to x.
- $(\mathrm{com}, \mathrm{st}) \leftarrow \mathrm{P}_1(\mathrm{PK})$: on input $\mathrm{PK} = (\mathbb{G}, q, g, h)$, it samples a uniform $k \in \mathbb{Z}_q^*$ and sets $\mathrm{com} := g^k$, $\mathrm{st} := k$.
- $\mathrm{rsp} \leftarrow \mathrm{P}_2(\mathrm{SK}, \mathrm{st}, \mathrm{ch})$: on input a private key x, a state k and a challenge ch, it returns $\mathrm{ch} \cdot x + k \pmod q$.
- $1/0 \leftarrow \mathrm{V}(\mathrm{PK}, \mathrm{com}, \mathrm{ch}, \mathrm{rsp})$: for the public key $\mathrm{PK} = (\mathbb{G}, q, g, h)$, it checks whether $g^{\mathrm{rsp}} \cdot h^{-\mathrm{ch}} = \mathrm{com}$.

Security of Schnorr Identification Scheme

Theorem

If the discrete logarithm problem is hard relative to $\mathcal{G}$, then I_{Sch} is secure against a passive attack.

Sketch proof.

Let $\mathcal{A}$ be a PPT adversary in the identification exeperiment.

Valid transcripts can be simulated from $(\mathbb{G}, q, g, h)$:

- sample uniform and independent $\mathrm{ch}^*, \mathrm{rsp}^* \in \mathbb{Z}_q$;
- set $\mathrm{com}^* := g^{\mathrm{rsp}^*} h^{-\mathrm{ch}^*}$.
- The transcript $(\mathrm{com}^*, \mathrm{ch}^*, \mathrm{rsp}^*)$ is indistinguishable from an honest one.

Learning an honest transcript $(\mathrm{com}, \mathrm{ch}, \mathrm{rsp})$ does not give any *new* information to $\mathcal{A}$.

Security of Schnorr Identification Scheme

Sketch proof.

If $\mathcal{A}$, given $h, \mathrm{com} \in \mathbb{G}$, can output a response for any challenge with high probability, then it can respond with correct responses $\mathrm{rsp}_1, \mathrm{rsp}_2$ to two distinct challenge values $\mathrm{ch}_1, \mathrm{ch}_2 \in \mathbb{Z}_q$.

Therefore, $\mathcal{A}$ implicitly knowns $\log_g h$, since:

$$g^{\mathrm{rsp}_1} \cdot h^{-\mathrm{ch}_1} = \mathrm{com} = g^{\mathrm{rsp}_2} \cdot h^{-\mathrm{ch}_2} \quad \Rightarrow \quad h = g^{\frac{\mathrm{rsp}_2 - \mathrm{rsp}_1}{\mathrm{ch}_2 - \mathrm{ch}_1}}.$$

$\mathcal{A}$ can be exploited as a subroutine to construct an adversary $\mathcal{A}'$ against the discrete logarithm problem.

The Schnorr Signature Scheme

The Schnorr Signature Scheme $(\text{KeyGen}', \text{Sign}, \text{Verify})$ is obtained applying the Fiat-Shamir transform to the Schnorr identification protocol.

- $(\text{PK}, \text{SK}) \leftarrow \text{KeyGen}'(n)$: it runs KeyGen on input a security parameter n, obtaining $(\mathbb{G}, q, g, h)$ and x. A hash function $H : \{0,1\}^* \rightarrow \Omega_{\text{ch}}$ is also specified.
  Then PK is set to $(\mathbb{G}, q, g, h, H)$ and SK is set to (x, H).

- $\sigma \leftarrow \text{Sign}(\text{SK}, m \in \{0,1\}^*)$: on input a secret key (x, H) and a message m, it samples a uniform $k \in \mathbb{Z}_q^*$, computes $\text{com} := g^k$, $\text{ch} := H(\text{com}, m)$ and $\text{rsp} := \text{ch} \cdot x + k \pmod q$, and returns $\sigma := (\text{com}, \text{ch}, \text{rsp})$.

- $1/0 \leftarrow \text{Verify}(\text{PK}, m, \sigma)$: given a public key $(\mathbb{G}, q, g, h, H)$, a message m and a signature $(\text{com}, \text{ch}, \text{rsp})$, it outputs 1 if $H(\text{com}, m) = \text{ch}$ and $g^{\text{rsp}} \cdot h^{-\text{ch}} = \text{com}$.

Security of the Schnorr Signature Scheme

Corollary

If the discrete logarithm problem is hard relative to $\mathcal{G}$ and H is modelled as a random oracle, then the Schnorr signature scheme is existentially unforgeable.

Digital Signature Algorithm - DSA/ECDSA

Some of its versions go back to 1991.

Both in the Digital Signature Standard (DSS) by NIST.

It is *based* on an identification protocol that is secure if the discrete logarithm problem is hard.

DSA/ECDSA Identification Scheme

The DSA/ECDSA identification scheme $I_{DSA} = (\text{KeyGen}, \text{P}_1, \text{P}_2, \text{V})$ with $\Omega_{\text{ch}} = \mathbb{Z}_q \times \mathbb{Z}_q$ is defined as follows.

- $(\text{PK}, \text{SK}) \leftarrow \text{KeyGen}(n)$: it runs a group generation algorithm $\mathcal{G}$ on a security parameter n, obtaining a cyclic group $\mathbb{G}$ - with $|\mathbb{G}| = q$ and $||q|| = n$ - and a generator g.

  It samples a uniform $x \in \mathbb{Z}_q$ and computes $h := g^x$. Then it sets PK to $(\mathbb{G}, q, g, h)$ and SK to x.
- $(\text{com}, \text{st}) \leftarrow \text{P}_1(\text{PK})$: on input $\text{PK} = (\mathbb{G}, q, g, h)$, it samples a uniform $k \in \mathbb{Z}_q^*$ and sets $\text{com} := g^k$, $\text{st} := k$.
- $\text{rsp} \leftarrow \text{P}_2(\text{SK}, \text{st}, \text{ch})$: on input a private key x, a state k and a challenge $\text{ch} = (c, \alpha)$, it returns $k^{-1}(\alpha + x \cdot c) \pmod{q}$.
- $1/0 \leftarrow \text{V}(\text{PK}, \text{com}, \text{ch}, \text{rsp})$: for the public key $\text{PK} = (\mathbb{G}, q, g, h)$, it checks whether rsp is different from 0 and $g^{\alpha \cdot \text{rsp}^{-1}} \cdot h^{c \cdot \text{rsp}^{-1}} = \text{com}$.

DSA/ECDSA Identification Scheme

Correctness: as long as $\mathrm{rsp} \neq 0$, i.e. $\alpha \neq -x \cdot c \pmod{q}$, which does not happen with negligible probability.

Security: it is based on the hardness of the discrete logarithm problem relative to $\mathcal{G}$.

- Transcripts can be simulated.
- If $(\mathrm{com}, (\alpha, c_1), \mathrm{rsp}_1)$ and $(\mathrm{com}, (\alpha, c_2), \mathrm{rsp}_2)$ are two valid transcripts, then

$$g^{\alpha \cdot \mathrm{rsp}_1{}^{-1}} \cdot h^{c_1 \cdot \mathrm{rsp}_1{}^{-1}} = g^{\alpha \cdot \mathrm{rsp}_2{}^{-1}} \cdot h^{c_2 \cdot \mathrm{rsp}_2^{-1}} \Rightarrow h = g^{\frac{\alpha(\mathrm{rsp}_2^{-1} - \mathrm{rsp}_1^{-1})}{c_1 \cdot \mathrm{rsp}_1^{-1} - c_2 \cdot \mathrm{rsp}_2^{-1}}}.$$

DSA/ECDSA

The Digital Signature Algorithm DSA/ECDSA (KeyGen′, Sign, Verify) is defined as follows.

- $(\mathrm{PK}, \mathrm{SK}) \leftarrow \mathrm{KeyGen}'(n)$: it runs the key-generation algorithm of I_{DSA} on input a security parameter n, obtaining $(\mathcal{G}, q, g, h)$ and x. Two functions, $H : \{0,1\}^* \to \mathbb{Z}_q$ and $F : \mathbb{G} \to \mathbb{Z}_q$, are also specified.
  Then PK is set to $(\mathbb{G}, q, g, h, H, F)$, SK is set to (x, H, F).
- $\sigma \leftarrow \mathrm{Sign}(\mathrm{SK}, m \in \{0,1\}^*)$: on input a secret key (x, H, F) and a message m, it chooses a uniform $k \in \mathbb{Z}_q^*$ and sets $\mathrm{com} := g^k$, $c := F(\mathrm{com})$ and $\alpha := H(m)$. If $c = 0$ or $\alpha = -x \cdot c \pmod q$, it starts again by choosing a fresh k. Otherwise, it returns $(\mathrm{com}, (c, \alpha), \mathrm{rsp} := k^{-1} \cdot (\alpha + x \cdot c) \pmod q)$.
- $1/0 \leftarrow \mathrm{Verify}(\mathrm{PK}, m, \sigma)$: for the public key $(\mathbb{G}, g, q, h, H, F)$ it checks whether $g^{H(m) \cdot \mathrm{rsp}^{-1}} \cdot h^{F(\mathrm{com}) \cdot \mathrm{rsp}^{-1}} = F(\mathrm{com})$.

Security of DSA/ECDSA

DSA/ECDSA can be proven secure assuming the hardness of the discrete logarithm problem relative to $\mathcal{G}$ and modelling H and F as random oracles.

No known proofs when F is specified as in the standard (F is a simple function, not intended to act as a random one).

Security of DSA/ECDSA

DSA/ECDSA can be proven secure assuming the hardness of the discrete logarithm problem relative to $\mathcal{G}$ and modelling H and F as random oracles.

No known proofs when F is specified as in the standard (F is a simple function, not intended to act as a random one).

Knowledge of k implies knowledge of the secret key.

Re-use of the same k leads to the private key as well. Hackers exploited this against Sony PS3 in 2010.

Further Reading I

Carlisle Adams and Steve Lloyd.
Understanding PKI: concepts, standards, and deployment considerations.
Addison-Wesley Professional, 2003.

Dan Boneh, Ben Lynn, and Hovav Shacham.
Short signatures from the Weil pairing.
Journal of cryptology, 17(4):297–319, 2004.

Tim Dierks.
The transport layer security (TLS) protocol version 1.2.
2008.

Carl Ellison and Bruce Schneier.
Ten risks of PKI: What you're not being told about public key infrastructure.
Comput Secur J, 16(1):1–7, 2000.

Further Reading II

Amos Fiat and Adi Shamir.
How to prove yourself: Practical solutions to identification and signature problems.
In Advances in Cryptology—CRYPTO'86, pages 186–194. Springer, 1987.

Martin Georgiev, Subodh Iyengar, Suman Jana, Rishita Anubhai, Dan Boneh, and Vitaly Shmatikov.
The most dangerous code in the world: validating SSL certificates in non-browser software.
In Proceedings of the 2012 ACM conference on Computer and communications security, pages 38–49. ACM, 2012.

Further Reading III

Hugo Krawczyk.
Cryptographic extraction and key derivation: The HKDF scheme.
In Annual Cryptology Conference, pages 631–648. Springer, 2010.

Hugo Krawczyk, Kenneth G Paterson, and Hoeteck Wee.
On the security of the TLS protocol: A systematic analysis.
In Advances in Cryptology–CRYPTO 2013, pages 429–448. Springer, 2013.

Leslie Lamport.
Constructing digital signatures from a one-way function.
Technical report, Technical Report CSL-98, SRI International Palo Alto, 1979.