New Static Analysis Techniques to Detect Entropy Failure Vulnerabilities

Andrew Russell & Rushi Shah

December 11, 2018

The University of Texas at Austin

Cryptography & Digital signature schemes

Digital Signature Schemes

- Canonical cryptographic protocols = encryption and decryption = *data secrecy*

Digital Signature Schemes

- Canonical cryptographic protocols = encryption and decryption = *data secrecy*
- Digital signature schemes = *data authenticity* = who sent the data

Scheme procedures

Signature generation

Signature generation produces a public key p (an identity) and a secret key (a signature) s

Signing

Signing takes a message m and produces a signed message m'

Verification

Verification determines if a message m and a signed message m' came from a particular person (designated by their public key p)

Real world

This is used billions of times per day

vs.

Brief recap

Digital signature schemes provide *authenticity*, not secrecy

Still very necessary for any good cryptosystem

Elliptic Curves

Defining elliptic curves

Definition
An elliptic curve over a field $\mathbb{F}$ consists of the points (x, y) given by the equation

$$y^2 = x^3 + ax + b$$

where $a, b \in \mathbb{F}$ and $x^3 + ax + b$ has no repeated factors.

(a field is a set where we can add, subtract, multiply and divide except for 0; think $\mathbb{R}$ or $\mathbb{Q}$)

Examples

The curve $a = -1$ and $b = 1$ over $\mathbb{R}$:

Examples

The curve $a = -1$ and $b = 1$ over $\mathbb{F}_{163}$:

Arithmetic

We can perform arithmetic on the points of the elliptic curve.
To add two points P and Q:

1. Draw the secant line between the two points

Arithmetic

We can perform arithmetic on the points of the elliptic curve.
To add two points P and Q:

1. Draw the secant line between the two points
2. Find the third point of intersection of this line and the curve

Arithmetic

We can perform arithmetic on the points of the elliptic curve. To add two points P and Q:

1. Draw the secant line between the two points
2. Find the third point of intersection of this line and the curve
3. Find the reflection of this point across the x-axis, and we're done!

Arithmetic example

Arithmetic example

Arithmetic example

Adding a point to itself

For an integer n, we denote P added to itself n times by nP

Adding a point to itself

For an integer n, we denote P added to itself n times by nP

Adding a point to itself

For an integer n, we denote P added to itself n times by nP

Fun fact

The points of an elliptic curve actually form an abelian *group*.

Bilinear pairings

We have a function $\phi : E \times E \to E$ such that it is *bilinear* (we can pull out coefficients from either argument):

$$\phi(aP, Q) = a \cdot \phi(P, Q)$$

and

$$\phi(P, bQ) = b \cdot \phi(P, Q)$$

Digital signatures using elliptic curves

Elliptic curves and a signature scheme

Some setup: a trusted third party will generate an elliptic curve E and a point $Q \in E$ randomly, and publish both publicly for everyone to see (e.g. via the Internet).

Signature generation

Simply generate a random integer $s \in \mathbb{Z}$ and compute the elliptic curve point $P = sQ$.

Your **secret key** is s and your **public key** is P.

Signing

Let $m \in \mathbb{Z}$ be a message.

Signing

Let $m \in \mathbb{Z}$ be a message. To sign this message, first compute the elliptic curve point $M = mQ$.

Thus, $M' = sM = s(mQ)$ is the signed message.

Verification

To verify, we compute $R_1 = \phi(Q, M')$ and $R_2 = \phi(P, mQ)$ and test if $R_1 = R_2$.

Verification

To verify, we compute $R_1 = \phi(Q, M')$ and $R_2 = \phi(P, mQ)$ and test if $R_1 = R_2$.

To see why this works: since $P = sQ$ and $M' = s(mQ)$ then by the bilinearity of ϕ we have:

$$\begin{aligned} R_1 &= \phi(Q, M') \\ &= \phi(Q, smQ) \\ &= sm \cdot \phi(Q, Q) \end{aligned}$$

Verification

To verify, we compute $R_1 = \phi(Q, M')$ and $R_2 = \phi(P, mQ)$ and test if $R_1 = R_2$.

To see why this works: since $P = sQ$ and $M' = s(mQ)$ then by the bilinearity of ϕ we have:

$$\begin{aligned} R_1 &= \phi(Q, M') \\ &= \phi(Q, smQ) \\ &= sm \cdot \phi(Q, Q) \end{aligned}$$

$$\begin{aligned} R_2 &= \phi(P, mQ) \\ &= \phi(sQ, mQ) \\ &= s \cdot \phi(Q, mQ) \\ &= sm \cdot \phi(Q, Q) \end{aligned}$$

Security?

Source: https://xkcd.com/538/

Attacking our scheme

Given just a public key $P = sQ$ and public point Q, if we could recover the secret key s then we can break the scheme by being able to sign any message we want

Security in modern cryptography

Modern crypto defines security through reductions to “hard problems”

Security in modern cryptography

Modern crypto defines security through reductions to “hard problems”

Proof of security: a cryptosystem X is *just as hard* to hack as it is to solve a “hard” computational problem

Security in modern cryptography

Modern crypto defines security through reductions to “hard problems”

Proof of security: a cryptosystem X is *just as hard* to hack as it is to solve a “hard” computational problem

Ex.: *If* you can factor large integers quickly, ***then*** you can break the RSA encryption scheme

Defining our hard problem

We can make an analogy to computing logarithms: $P = Q^s$ and Q

(because the points of an elliptic curve form a group, our $sQ = Q + Q + \cdots + Q$ could be written multiplicatively as $Q^s = Q \cdot Q \cdots Q$)

Defining our hard problem

We can make an analogy to computing logarithms: $P = Q^s$ and Q

(because the points of an elliptic curve form a group, our $sQ = Q + Q + \cdots + Q$ could be written multiplicatively as $Q^s = Q \cdot Q \cdots Q$)

So, we want to take the "logarithm" of both sides to recover s:

$$\log_Q(P) = s$$

This is easy, with say, $\mathbb{F} = \mathbb{R}$

Discrete log problems

This is easy, with say, $\mathbb{F} = \mathbb{R}$

It is very hard in a discrete setting, with $\mathbb{F} = \mathbb{Z}/p\mathbb{Z}$

Discrete log problems

This is easy, with say, $\mathbb{F} = \mathbb{R}$

It is very hard in a discrete setting, with $\mathbb{F} = \mathbb{Z}/p\mathbb{Z}$

This is known as the discrete log problem

The discrete log problem

Discrete Log Problem: given two elliptic curve points $R = xP$ and P over a *finite field*, we cannot easily compute $x \in \mathbb{Z}$

Recap

We've covered:

- Basic elliptic curve arithmetic
- Digital signature schemes from EC machinery (particularly, pairing)
- A canonical "proof of security" for why this scheme is secure

My thesis is primarily on the *pairing* portion (specifically, the Weil pairing)

Questions?

and thank you!