The Mathematics of Digital Signatures

Angela Robinson

Handwritten signatures have been used to verify the authenticity of documents for centuries. In the late 1970s, mathematicians discovered pivotal techniques to construct digital signatures for authenticating any message or data that can be represented as bit strings [DH76, RSA78]. Digital signatures quickly evolved from theoretical tools to essential components of our global digital world. Software patches and updates are digitally signed by the provider so that before a device installs the update, the device can verify the origin of the software and that the software package was not modified after the signature was applied. Transactions on public blockchains like Bitcoin and Ethereum are digitally signed by the coin-sender to authenticate the details of the transaction (coin amount to be sent, recipient, etc.) and to prevent unauthorized changes to the transaction details.

Components of a digital signature. Digital signature algorithms (DSAs) are used to establish authenticity and integrity. The former assures that the signed message originated from the claimed sender and the latter assures that the message has not been changed during transit. A typical example is illustrated in Figure 1.

Angela Robinson is a mathematician at the National Institute of Standards and Technology. Her email address is angela.robinson@nist.gov.

Communicated by Notices *Associate Editor Emilie Purvine.*

DOI: https://doi.org/10.1090/noti2722

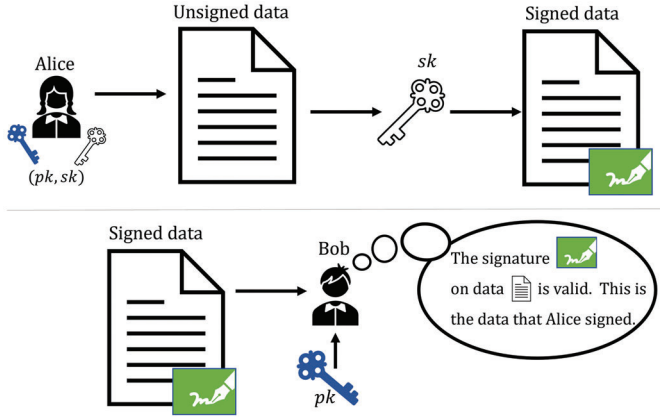

Figure 1. Alice generates a pair of keys and sends a signed message to Bob. Bob uses Alice's public key to verify the signature.

There are three components of a digital signature scheme: `key generation`, `sign`, `verify`.

During the first phase of the protocol, `key generation` generates a pair of keys: one public key *pk* for signature verification and one secret key *sk* for signing messages. To prevent forgeries, it is critical that the signer does not share *sk* with anyone.

The `sign` algorithm can use *sk* to sign any message or data. Unlike handwritten signatures, a digital signature depends on the message and will thus vary. Upon receipt of a signed message, the message recipient uses public knowledge of *pk* to run the `verify` algorithm to determine whether the signature is valid with respect to the message.

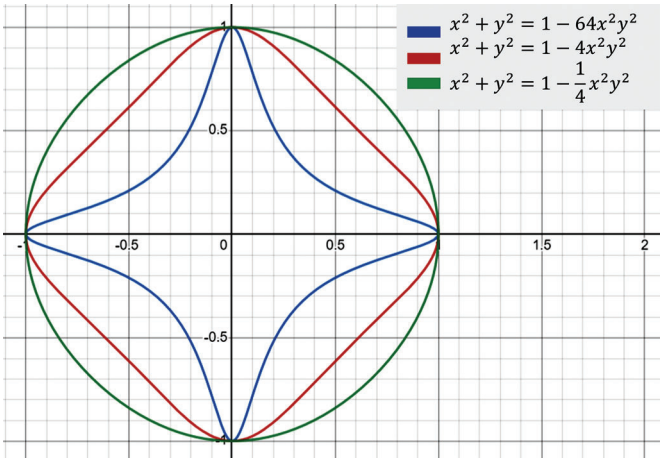

Figure 2. Edwards curve examples.

Constructing digital signatures. Many digital signature schemes are constructed using *trapdoor* functions. That is, functions that are easy to compute but difficult to invert without some knowledge of the trapdoor. One such digital signature scheme is the Edwards-curve Digital Signature Algorithm (EdDSA).

EdDSA belongs to a family of digital signature algorithms that use elliptic curves over finite fields. It is well known that for an elliptic curve $\mathcal{E}$ defined over a finite field $\mathbb{F}_q$, $\mathcal{E}(\mathbb{F}_q)$ forms an abelian group under addition with respect to a particular addition law. At a high level, an Edwards curve $\mathcal{E}$ is defined by $x^2 + y^2 = 1 + dx^2y^2$ over $\mathbb{F}_q$, $d \neq 0, 1$ [Edw07]. Some examples of Edwards curves over $\mathbb{R}$ with various d values are given in Figure 2. The Edwards addition law is, for two points $(x_1, y_1), (x_2, y_2) \in \mathcal{E}(\mathbb{F}_q)$,

$$(x_1, y_1) + (x_2, y_2) = \left(\frac{x_1y_2 + x_2y_1}{1 + dx_1x_2y_1y_2}, \frac{y_1y_2 - x_1x_2}{1 - dx_1x_2y_1y_2} \right).$$

The security of EdDSA is based on the difficulty of solving the Elliptic Curve Discrete Logarithm Problem (ECDLP). Let $P \in \mathcal{E}(\mathbb{F}_q)$ be a point of prime order p, and let $\langle P \rangle$ be the subgroup generated by P. For any $Q \in \langle P \rangle$, $Q = a \cdot P$ for some $a \in [0, p-1]$. The ECDLP is, given P, Q, and $\mathcal{E}$, find a. Computing $a \cdot P$ is computationally easy, but solving the ECDLP requires significantly more computation time. In fact, for appropriately chosen parameters, solving the ECDLP is computationally infeasible.

EdDSA. The following is a highly simplified version of how the components of EdDSA are used to sign a message m using an Edwards curve $\mathcal{E}(\mathbb{F}_q)$. Let H be a function that maps messages of arbitrary length to bit strings of a fixed size in a manner that is difficult to invert and difficult to find a pair of distinct input strings that map to the same output.

`key generation`: Selects a point $P \in \mathcal{E}(\mathbb{F}_q)$ of prime order p, and a random integer $a \in [1, p-1]$. The point P is a public parameter, pk is the point $Q = a \cdot P = (x, y)$, and sk is the scalar a.

`sign`: Given message m and a, computes integer $r = H(H(a) + m) \mod q$ and point $R = r \cdot P$. For simplicity, assume that m and points R, Q are encoded in such a way that enables addition modulo q. Let $h = H(R + Q + m) \mod q$. The signature on m is the pair $(R, s) = (R, r + h \cdot a \mod q)$.

`verify`: Computes $h = H(R + Q + m) \mod q$ and two points: $P_1 = s \cdot P$, and $P_2 = R + h \cdot Q$. If $P_1 = P_2$, the signature (R, s) is accepted. Otherwise, rejected. The curious reader is encouraged to check that `verify` accepts well-formed signatures.

Integrity is broken if an attacker manages to forge a signature on a new message m'. It is not known how to forge EdDSA (at cryptographic sizes) signatures without knowledge of sk, and recovering sk from pk requires one to solve the ECDLP. The security of EdDSA depends on the difficulty of solving the ECDLP.

The predecessor of EdDSA, the elliptic curve digital signature algorithm (ECDSA), emerged in the 1990s and performed remarkably better than the DSAs of the 1970s. EdDSA varies a bit from ECDSA but can provide even more efficiency with additional security protections. EdDSA has recently been included in US cryptographic standards [Nat23].

References

[DH76] Whitfield Diffie and Martin E. Hellman, *New directions in cryptography*, IEEE Trans. Inform. Theory **IT-22** (1976), no. 6, 644–654, DOI 10.1109/tit.1976.1055638. MR437208

[Edw07] Harold M. Edwards, *A normal form for elliptic curves*, Bull. Amer. Math. Soc. (N.S.) **44** (2007), no. 3, 393–422, DOI 10.1090/S0273-0979-07-01153-6. MR2318157

[Nat23] National Institute of Standards and Technology, *Digital signature standard (DSS)*, Technical Report Federal Information Processing Standards (FIPS) Publication 186-5, U.S. Department of Commerce, Washington, D.C., 2023.

[RSA78] R. L. Rivest, A. Shamir, and L. Adleman, *A method for obtaining digital signatures and public-key cryptosystems*, Comm. ACM **21** (1978), no. 2, 120–126, DOI 10.1145/359340.359342. MR700103

Angela Robinson

Credits

Figure 1 and Figure 2 are courtesy of Angela Robinson. Photo of Angela Robinson is courtesy of Keyi Liu.