# 2025–40 years of ECC

## Implementations, impact, and extensions

Kristin Lauter
Meta AI Research
August 11, 2025

# Commercial deployment at Microsoft

# ECC at Microsoft Research (1999-2005)

Peter Montgomery:

- Bignum library (multi-precision arithmetic library)
- Montgomery curves, y^2 = x^3 + a*x^2 + x
  - e.g. ECC25519 over Fp, p = 2^{255} - 19, pseudo-Mersenne, introduced by Dan Bernstein (2005)
  - Widely deployed in many libraries and protocols, using Montgomery ladder
- Montgomery multiplication

## Coordinate systems: affine vs. (weighted) projective

- For pseudo Mersenne primes, reported ratio of the cost of inversion to multiplication modulo p: **80-to-1**
- For random primes, with Peter's fast inversion techniques, ratio was ~ **5-to-1**
- Simultaneous inversion trick
- → implementation of ECC and pairing-based crypto with affine coordinates → numerous improvements
  - An Efficient Procedure to Double and Add Points on an Elliptic Curve, Eisentraeger-Lauter-Montgomery (2002/112)
  - Fast Elliptic Curve Arithmetic and Improved Weil Pairing Evaluation, Eisentraeger-Lauter-Montgomery (2003/242)
  - Trading Inversions for Multiplications in Elliptic Curve Cryptography, Ciet-Joye-Lauter-Montgomery (2003/257)
  - An Analysis of Affine Coordinates for Pairing Computation, Lauter-Montgomery-Naehrig (2010/363)

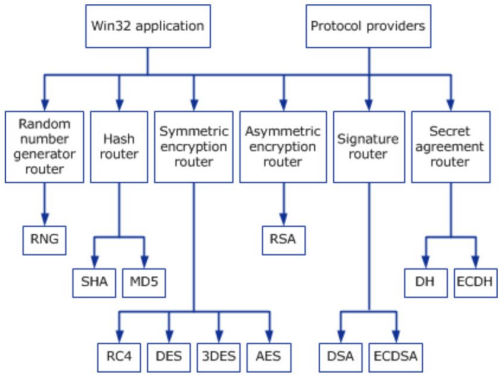
# ECC in Windows Vista CNG (2005)

CNG provides primitives for the following classes of algorithms.

Expand table

| Algorithm class | Description |
|---|---|
| Random number generator | Pluggable random number generation (RNG). |
| Hashing | Algorithms used for hashing, such as SHA1 and SHA2. |
| Symmetric encryption | Algorithms used for symmetric encryption, such as AES, 3DES, and RC4. |
| Asymmetric encryption | Asymmetric (public key) algorithms that support encryption, such as RSA. |
| Signature | Signature algorithms such as DSA and ECDSA. This class can also be used with RSA. |
| Secret agreement | Secret agreement algorithms such as Diffie-Hellman (DH) and elliptic curve Diffie-Hellman (ECDH). |

The following illustration shows the design and function of the CNG cryptographic primitives.




# NSA Suite B Crypto, Feb 2005

NSA Suite B contains the following algorithms:[5]

| Purpose | Algorithms | Parameter Length | |
|---|---|---|---|
| | | Secret | Top-Secret |
| Encryption | AES | 128 | 256 |
| Digital Signature | Elliptic Curve Digital Signature Algorithm (ECDSA) | 256 | 384 |
| Key Exchange | Elliptic-curve Diffie–Hellman (ECDH) | 256 | 384 |
| Hash Function | SHA-2 | 256 | 384 |

https://learn.microsoft.com/en-us/windows/win32/seccng/cryptographic-primitives?source=recommendations

# 2005: Windows Vista Elliptic Curve Cryptography CNG

**AI Overview:** Microsoft Windows Vista introduced the Cryptography API: Next Generation (CNG), which includes support for Elliptic Curve Cryptography (ECC). CNG provides a unified set of APIs for both user and kernel mode cryptography, enabling the use of ECC and other modern cryptographic algorithms in various applications, including SSL/TLS and IPsec.

Key points about ECC and CNG in Windows Vista:

> **CNG's Role:** CNG was designed as a replacement for the older Cryptographic API (CAPI) offering greater flexibility and support for new algorithms like ECC.
> **ECC Implementation:** CNG provides implementations of ECC algorithms within its key storage and retrieval functions, specifically within the NCrypt subset.

In simpler terms: Windows Vista's CNG API made it easier to use modern cryptography, including ECC, in various applications, including those requiring kernel-level security features like SSL/TLS and IPsec. It also improved smart card support and provided a more flexible and extensible framework for cryptography in Windows.

# Windows CE deployment–OS for embedded devices

**AI Overview:** Windows CE supports ECC through the .NET Compact Framework and third-party libraries or custom implementations. Developers can leverage these options to integrate ECC for secure communication and data protection within their Windows CE applications.

**Popular Frameworks and Tools for ECC-256**

- **Bouncy Castle:** cryptography library for .NET that supports ECC
- **Microsoft .NET Framework:** Provides built-in support for ECC through classes in the System.Security.Cryptography namespace.
- **OpenSSL:** OpenSSL can be integrated with C# applications for ECC operations. X25519 is an elliptic curve Diffie-Hellman key exchange using Curve25519, which was added to `openssl 1.1.0`, and Red Hat Enterprise Linux 8 supports up to `openssl-1.1.1`. (Red Hat Customer Portal)

# Pairing-based Applications

# Signatures for Network Coding

Uses ECC-based BLS signatures to prevent pollution attacks in content distributions networks like BitTorrent

## Homomorphic signatures for network coding

1 language

Article Talk Read Edit View history Tools

From Wikipedia, the free encyclopedia

Network coding has been shown to optimally use bandwidth in a network, maximizing information flow but the scheme is very inherently vulnerable to pollution attacks by malicious nodes in the network. A node injecting garbage can quickly affect many receivers. The pollution of network packets spreads quickly since the output of (even an) honest node is corrupted if at least one of the incoming packets is corrupted.

An attacker can easily corrupt a packet even if it is encrypted by either forging the signature or by producing a collision under the hash function. This will give an attacker access to the packets and the ability to corrupt them. Denis Charles, Kamal Jain and Kristin Lauter designed a new homomorphic encryption signature scheme for use with network coding to prevent pollution attacks.[1]

Signatures for Network Coding, Charles-Jain-Lauter, Proceedings of fortieth annual Conference on Information Sciences and Systems, 2006

# An Anonymous Health Care System

Groth-Sahai Noninteractive zero-knowledge proof systems (NIZK) uses pairing-based elliptic curve cryptography to enable for example, ***Delegatable Anonymous Credentials,***

Eprint 2008/428 *Belenkiy-Camenisch-Chase-Kohlweiss-Lysyanskaya-Shacham*

Can be applied to enable insurance billing system which does not require the patient to share their medical record and procedures with the insurance company!

An Anonymous Health Care System, Melissa Chase, Kristin Lauter
USENIX 2010 HealthSec Workshop, *IACR Cryptology ePrint Archive* | January 2011 , Vol 16

# Elliptic Curves in Post Quantum Era

# The Quantum threat:

Polynomial time Quantum algorithms for attacking current systems!

m = # bits

- Shor's algorithm for factoring $4m^3$ time and 2m qbits
- ECC attack requires $360m^3$ time and 6m qbits

  [Proos-Zalka, 2004]

Conclusion:

- RSA: m = 2048
- Discrete log m = 2048
- Elliptic Curve Cryptography m = 256 or 384

*are not resistant to quantum attacks once a quantum computer exists at scale!*

# Timeline for Elliptic Curve Cryptography

- (2005)  Suite B set requirements for the use of Elliptic Curve Cryptography
- (2016) CNSA requirements increase the minimum bit-length for ECC from 256 to 384.  Advises that adoption of  ECC not required.
- (2017) NIST international competition to select post-quantum solutions: 5-year PQC Competition
- (2022) PQC algorithms standardized

# Post-quantum cryptography

Submissions to the NIST PQC competition based on hard math problems:

- Code-based cryptography (McEliece 1978)
- Multivariate cryptographic systems (Matsumoto-Imai, 1988)
- Lattice-based cryptography (Hoffstein-Pipher-Silverman, NTRU 1996)
- Supersingular Isogeny Graphs (Charles-Goren-Lauter 2005)

- Challenge!  Need to see if these new systems are resistant to *both* classical and quantum algorithms!

# *Supersingular Isogeny Graphs*

New hard problem introduced at NIST Hash Function Competition in 2005: [Charles-Goren-Lauter]

- *Finding paths between nodes in a Supersingular Isogeny Graph*

Cryptographic Application: collision-resistant hash functions

Graphs: G = (V, E) = (vertices, edges)

- k-regular, undirected graphs, with optimal expansion
- No known efficient routing algorithm

# Cryptographic hash functions from expander graphs

Denis Charles, Microsoft Research
Eyal Goren, McGill University
Kristin Lauter, Microsoft Research

ECC 2006, Fields Institute
September 18, 2006

# Practical applications

- Password verification
- Integrity check of received content
- Signed hashes
- Encryption protocols
- Message digest
- Microsoft source code (720 uses of MD5)

# Example: graph of supersingular elliptic curves modulo p (Pizer)

- Vertices: supersingular elliptic curves mod p
- Curves are defined over GF($p^2$)
- Labeled by j-invariants
- Vertices can also be thought of as maximal orders in a quaternion algebra
- # vertices ~ p/12
- p ~ $2^{256}$

# Pizer graph

- Edges: degree $\ell$ isogenies between them
- $k = \ell+1$ – regular
- Graph is Ramanujan (Eichler, Shimura)
- Undirected if we assume p == 1 mod 12

# Application: Cryptographic Hash functions

A *hash function* maps bit strings of some finite length to bit strings of some fixed finite length

$$h : \{0,1\}^n \rightarrow \{0,1\}^m$$

- easy to compute
- unkeyed (do not require a secret key to compute output)
- Collision resistant
- Uniformly distributed output

# Collision-resistance

- A hash function *h* is *collision resistant* if it is computationally infeasible to find two distinct inputs, *x, y*, which hash to the same output
  $h(x) = h(y)$
- A hash function h is *preimage resistant* if, given any output of h, it is computationally infeasible to find an input, x, which hashes to that output.

# Application: cryptographic hash function

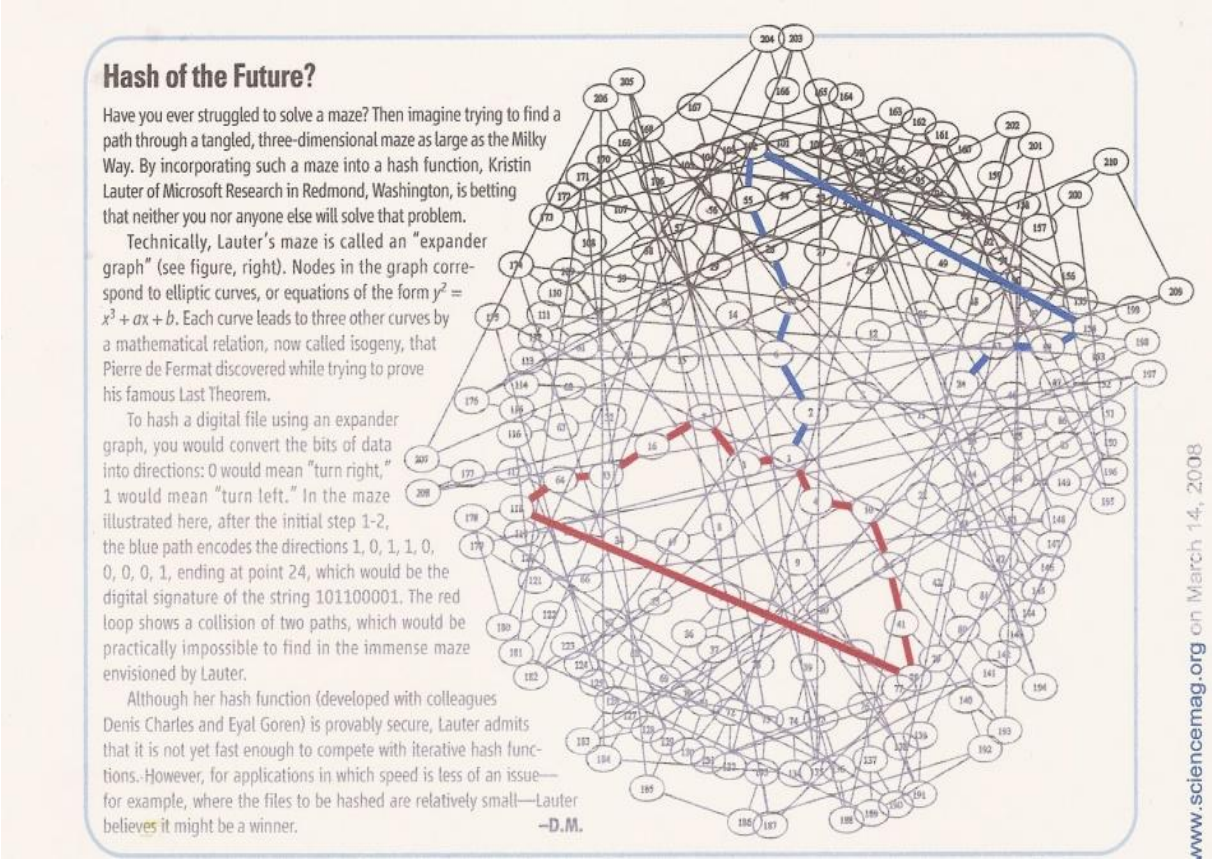
- k-regular graph G
- Each vertex in the graph has a label

**Input: a bit string**

- Bit string is divided into blocks
- Each block used to determine which edge to follow for the next step in the graph
- No backtracking allowed!

**Output: label of the final vertex of the walk**

# Science magazine 2008



## Hash of the Future?

Have you ever struggled to solve a maze? Then imagine trying to find a path through a tangled, three-dimensional maze as large as the Milky Way. By incorporating such a maze into a hash function, Kristin Lauter of Microsoft Research in Redmond, Washington, is betting that neither you nor anyone else will solve that problem.

Technically, Lauter's maze is called an "expander graph" (see figure, right). Nodes in the graph correspond to elliptic curves, or equations of the form $y^2 = x^3 + ax + b$. Each curve leads to three other curves by a mathematical relation, now called isogeny, that Pierre de Fermat discovered while trying to prove his famous Last Theorem.

To hash a digital file using an expander graph, you would convert the bits of data into directions: 0 would mean "turn right," 1 would mean "turn left." In the maze illustrated here, after the initial step 1-2, the blue path encodes the directions 1, 0, 1, 1, 0, 0, 0, 0, 1, ending at point 24, which would be the digital signature of the string 101100001. The red loop shows a collision of two paths, which would be practically impossible to find in the immense maze envisioned by Lauter.

Although her hash function (developed with colleagues Denis Charles and Eyal Goren) is provably secure, Lauter admits that it is not yet fast enough to compete with iterative hash functions. However, for applications in which speed is less of an issue—for example, where the files to be hashed are relatively small—Lauter believes it might be a winner. –D.M.

# Other graphs

- Vary the isogeny degree
- Lubotzky-Phillips-Sarnak graph
  - Cycles found: Eurocrypt 2008, Zemor-Tillich
  - Preimages found: SCN 2008, Petit-Quisquater-Lauter
    - LPS “path-finding” now used for quantum arithmetic (aka Ross-Selinger)
- Morgenstern graph, [Petit-Quisquater-Lauter 08]
- Genus 2 and Higher dimensional analogues
  - Superspecial abelian surfaces [Charles-Goren-L 07]
- Add level structure: [Arpin’22]

Adding Level Structure to Supersingular Elliptic Curve Isogeny Graphs

# “Isogenies in Cryptography” ongoing work:

- Alternate graphs/protocols:
  - CSIDH: Castryck-Lange-Martindale-Panny-Renes
- Dimension 2 analogues:
  - Decru, Flynn, Wesolowski, Jetchev, Florit, Smith …
- Signatures:
  - Vercauteren et al., Beullens,…
- Attacks:
  - Petit, Biasse, Bernstein, Stange, Morrison, …
- Graph structure:
  - Kohel, Arpin et al.