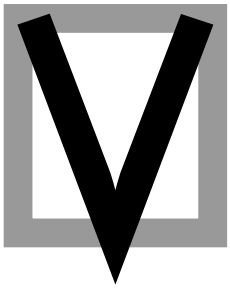
Verificatum Mix-Net

Technical fact sheet
2016-07-07



Secure and Flexible

- **Implementation of provably secure mix-net.**
  Completely faithful to cryptographic theory.
  Implemented by world-leading expert.
- **Unprecedented and unique blackbox functionality:**
  Distributed threshold key generation.
  Jointly decrypts ciphertexts.
  Shuffles ciphertexts using any public key.
  Mixes ciphertexts (shuffles + decrypts).
- **Everything is parameterized.**
  Any width of keys and ciphertexts.
  Security parameters, groups, communication model,...

Efficient

- **Runs in constant memory and scales linearly with:**
  servers, ciphertexts, width of ciphertexts
- **Example:**
  3 servers on 700 euro computers
  remotely located over plain Internet
  2/3 threshold for decryption
  P-256 elliptic curve (30 year security)
  1,000,000 ciphertexts
  Shuffles and decrypts in less than 30 minutes.
  Includes mutual verification of all ZK proofs.

Usage

- **Easy to use.**
  Basic use: simple commands with conservative defaults
  Complex use: Powerful optional parameters.
- **Mature and complete documentation.**
  High quality manual with examples.
  Detailed usage information for all commands.
  Commented generated configuration files.
- **Easy to demo both locally and remotely.**
  Bundled configurable demo script.
  Allows real-world remote demo.

Deployment

- **Easy to build and install.**
  Standard GNU build and install from source.
  Binary packages for major Linux distributions.
- **Portable.**
  POSIX and only one dependency.
  Runs on wide array of Unix/Linux systems.

Integration

- **Philosophy.**

  This is the Unix philosophy: Write programs that do one thing and do it well. Write programs to work together. Write programs to handle text streams, because that is a universal interface. – McIlroy

- **Dataformats.**

  Simplistic binary format for data with inspection tool.

  XML configuration files.

- **Interfaces.**

  Command line interface for most cases.

  Software interface for custom applications.

Universally Verifiable Proofs

Each execution generates a universally verifiable proof.

Mature, repeatedly revised, document describes proof. Level of detail is equivalent to NIST standard documents.

Undergrads at Tel Aviv University and KTH Royal Institute of Technology have implemented verifiers using the document with almost no advice.

Built-in verifier can print any subset of test vectors to help debug independently implemented verifiers.

Code Quality

- **Excellent code quality:**

  Project started 2005. Slow careful development.

  Everything is commented and documented.

  Static analyzers – CheckStyle, FindBugs, PMD

  Independent code analysis – Crisp AB, Joe Kiniry, and Demtech, Copenhagen Technical University (CTU)

- **Only two sources of external code:**

  GNU Multiple Precision Arithmetic Library (depency)

  Three OpenSSL curves (bundled, not using the library!)

"...superior in quality, both in the big-picture design and the implementation details than any other crypto library I have ever audited." (out of 35+ libraries)

— Joe Kiniry

Optimized

Pre-computes when possible.

Fast arithmetic using all known relevant techniques.

Exploits multiple cores.

Exploits parallelism of mix-servers where possible.

Abstraction framework prevents code cluttering.

Advice from developers of GMP and Oracle JVM!

Used in Real Elections

- **Used by Wombat in Israel.**

  Student union elections of Tel Aviv University (twice).

  Election of Meretz party leader (Meretz is a party of Israeli parliament).

- **Used by Agora Voting in Spain.**

  Several primary elections, municipal elections, etc, adding to at least 600,000 cast votes.

- **Used by Scytl in Norway**

  2013 Norwegian electronic election

  approximately 72,000 voters out of 250,000 eligible

Contact Information

Verificatum AB
Erik Dahlbergsgatan 51
115 57 Stockholm
Sweden

Email: info@verificatum.com
Phone: +46 73 687 0060

www.verificatum.com