ECC Design Team: A Second Report

Dan Brown, Russ Housley, Tim Polk, Sean Turner, Kelvin Yiu

December, 2007

Specifying ECC Public Keys

- RFC 3279
  - Algorithm OID indicates elliptic curve, and includes algorithm parameters
  - In conjunction with key usage extension, can restrict a key to signatures or key agreement
  - Cannot differentiate a key intended for DH from an MQV key

Design Team's Initial Proposal (from "final" report)

- Retain 3279 OID/parameters
  - Critical mass is finally emerging!
- Specify certificate extension as SHOULD implement for CAs and clients
  - Criticality provides opt-in/opt-out mechanism to select interoperability or control
  - Applications can take advantage of hints in noncritical extension, even where unrecognized by the path validation module
- Consistent with current application/protocol expectations (Algorithm OID plus extensions)

WG Response to Initial Proposal

- Don't put algorithm constraints in an extension.

ECC Design Team, Part 2

- Reformed Design Team
  - Decided two constraints needed to be supported for IETF protocols: *only DH* or *only MQV*
    - Constraints on hash algorithms (for signature keys) or KDFs (for DH and MQV) should be negotiated by the protocol
  - The ecPublicKey OID is mandatory to implement for IETF protocols
    - *Implementations* may be configured to require the constrained keys

Notes on ECC Signature Keys

- Key Usage already constrains usage
- Signature keys are inherently different
  - The signature verifier must use the algorithm and parameters specified by the signer to verify a signature, there is little chance for unintentional misuse of the public key.
- So, ecPublicKey is believed sufficient

Considered two strategies

- X9.62-2005 based
  - Restrictions are specified in the algorithm parameters in a SEQUENCE
  - IETF profile would limit SEQUENCE to only one restriction
- RFC 4055 based
  - Define two new algorithm OIDs, ecMQVPublicKey and ecDHPublicKey

X9.62-2005 based solution

- Pros
  - Strong alignment with ANSI and SECG
  - Migration path to additional granularity
  - Streamlined algorithm negotiation
- Cons
  - Application level parameter processing

RFC 4055 based Solution

- Pros
  - Same parameter structure for restricted and unrestricted public keys
  - No application level parameter processing
- Cons
  - No migration path to restrictions with higher granularity

Selected Proposal

- RFC 4055 based solution
  - Specify two new algorithm OIDs in X9 arc for inclusion in PKIX spec and X9.63
    - Retain the ecPublicKey algorithm syntax
  - IETF protocols that support the new OIDS MUST also support ecPublicKey

Rationale

- Protect deployed base for ECC keys
- Applications process same information for all ECC keys
- Compliant subset of X9 standards

Supplementary Design Team Proposals

- ECC Parameter handling
  - Named curves are more efficient to process than inherited parameters
  - MUST support for named curves
  - Support for explicit and inherited parameters is optional
- RFC 4055
  - KDF restrictions MUST not appear in certificates (currently SHOULD NOT)

Next Steps

- Design team will submit a new ID for consideration by WG
  - ID would obsolete both 3279 and 4055

Questions?