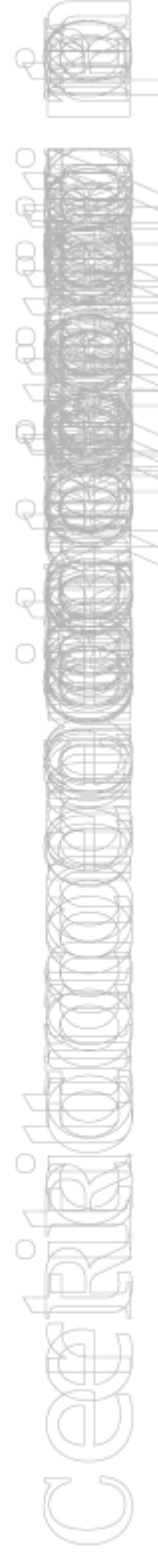
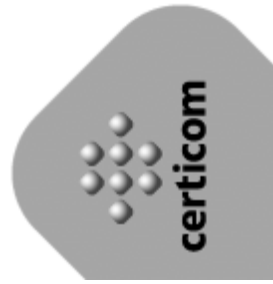


# ANSI X9.63 Overview

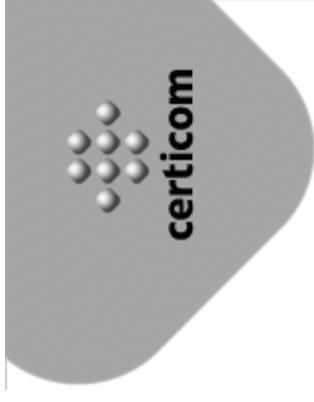
Key Agreement and Key Transport  
Using Elliptic Curve Cryptography

Simon Blake-Wilson  
Certicom



# Overview

- What is ANSI X9.63?
- Overview of ANSI X9.63
  - ♦ Goals
  - ♦ Primitives
  - ♦ Schemes
- Are schemes in ANSI X9.63 suitable for NIST?
- Conclusion



## What is ANSI X9.63?

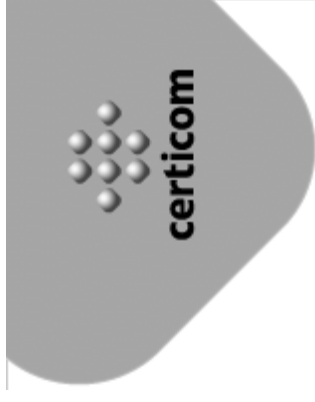
- Standard being developed by ANSI X9F1
- Primarily designed to meet the needs of the financial services industry, but also generally applicable
- Specifies key agreement and key transport schemes using elliptic curve cryptography
- Specifies a variety of schemes to meet the diverse security needs of communications protocols
- Specifies mathematical primitives needed to completely implement schemes
- Ballot due in near future



## ANSI X9.63 - Goals

Specify schemes capable of meeting common security needs. Security goals identified by the cryptographic community include:

- Implicit key authentication (IKA)
- Explicit key authentication (EKA)
- Entity authentication (EA)
- Known-key security (K-KS)
- Forward secrecy (FS)
- Key-compromise impersonation (K-CI)
- Etc

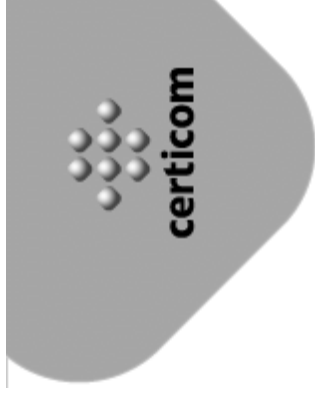


## ANSI X9.63 - Goals (2)

Specify schemes that as far as possible provide additional desirable attributes like:

- Minimal number of passes
- Low communication overhead
- Low computation overhead
- Limited reliance on timestamping
- Limited reliance on hash functions
- Etc

Efficiency goals particularly important for schemes using elliptic curve cryptography.

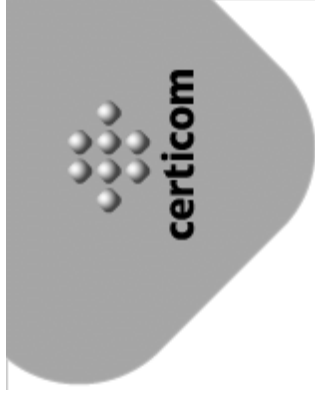


# ANSI X9.63 - Primitives

A number of primitives (mathematical building blocks) must be specified in order to build schemes.

Domain parameter generation & validation primitives:

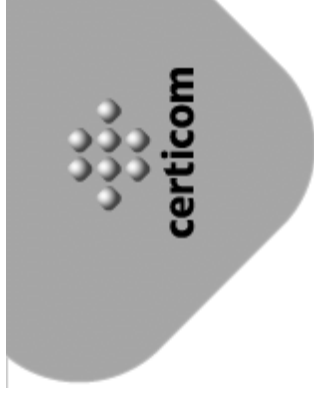
- Fp and  $F_{2^m}$  fields allowed
- m prime to avoid Weil descent
- Polynomial and normal bases allowed for  $F_{2^m}$
- $2^{160}$  minimum size
- Curves selected in any manner. Verifiably random selection option
- Mandatory to avoid known attacks
- List of recommended parameters provided



## ANSI X9.63 - Primitives (2)

Key generation & validation primitives:

- Secret keys selected randomly or pseudorandomly
- Key pairs bound to domain parameters
- Public key validation checks mathematical properties of key to ensure it's plausible
- Public key validation necessary in many circumstances to avoid attacks like small subgroup attacks
- Public key validated or partial validation specified
- Usually required to 'receive assurance' of validity of key



## ANSI X9.63 - Primitives (3)

Diffie-Hellman primitives:

- Standard and cofactor Diffie-Hellman primitives specified
- Cofactor Diffie-Hellman provides efficient resistance to some attacks

MQV primitive:

- Derives shared secret value from two key pairs from each entity
- Particularly efficient manner to provide some security services



## ANSI X9.63 - Primitives (4)



Key derivation function:

- Derives keying material from shared secret value
- Simple hash function construction specified
- ASN.1 encoding may or may not be used

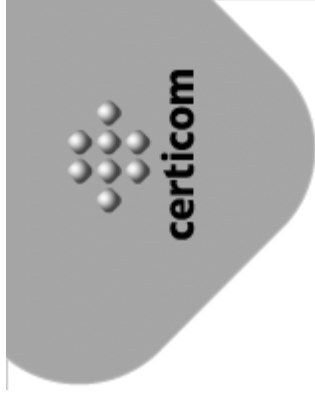
Asymmetric encryption scheme:

- Bellare and Rogaway's ECIES
- Based on elliptic curve Diffie-Hellman, XOR encryption, and a MAC
- Efficient, "provably secure" way to provide desirable security properties

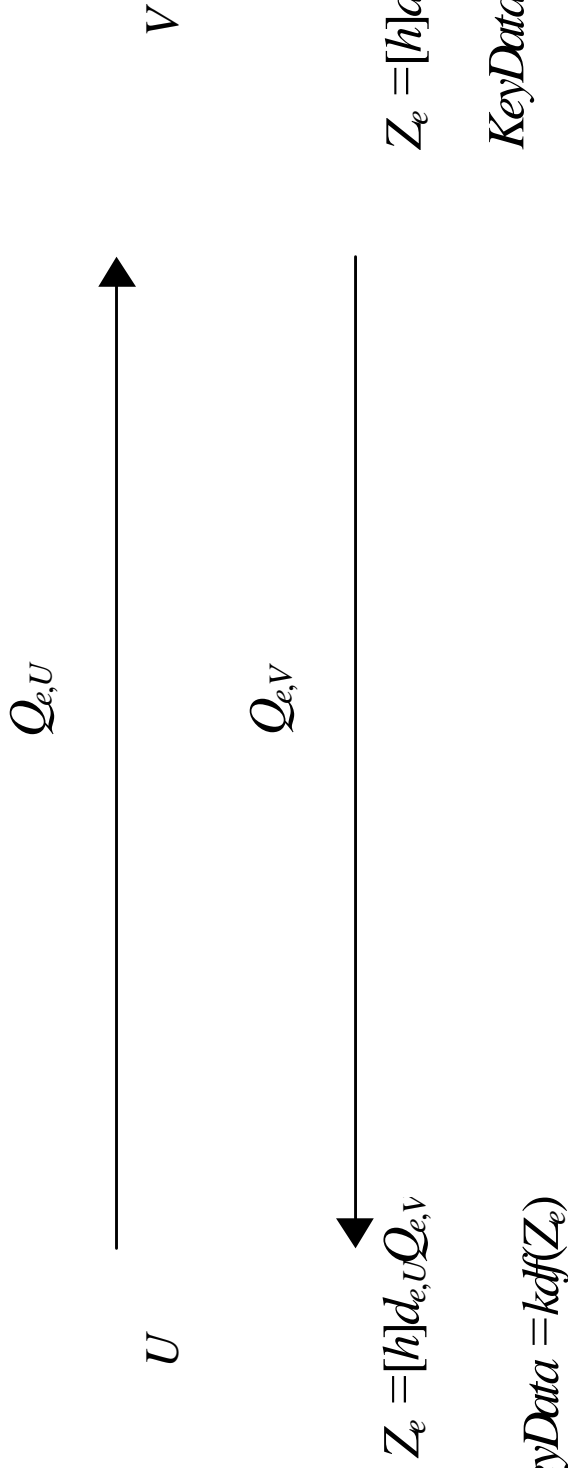
## **ANSI X9.63 - Primitives (5)**

In addition, a number of other building are required. These are “borrowed” from other ANSI standards:

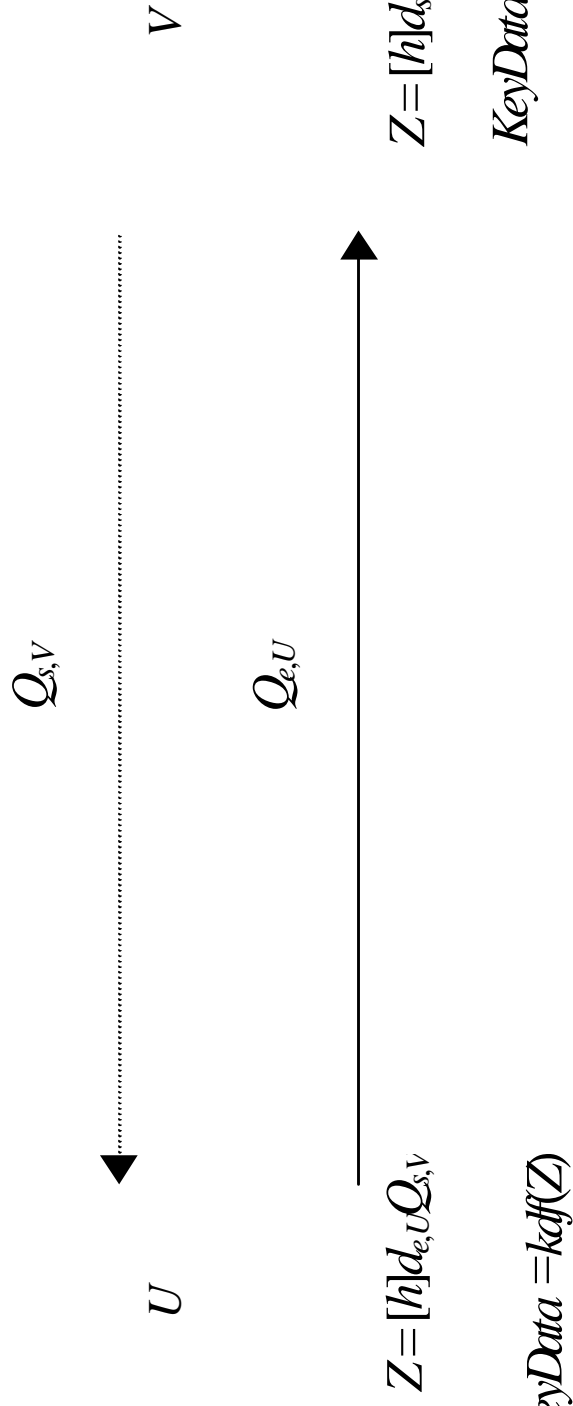
- ANSI-approved hash function (SHA-1)
- ANSI-approved random and pseudorandom number generators
- ANSI-approved MAC (HMAC with SHA-1)
- ECDSA



# ANSI X9.63 - Ephemeral UMI



# ANSI X9.63 - 1-Pass DH



# ANSI X9.63 - Static UM



$Q_{s,U}$



$U$

$V$

$Q_{s,V}$



$Z_s = [h]d_{s,U}Q_{s,V}$

$Z_s = [h]d_{s,V}Q_{s,U}$

$KeyData = kdf(Z_s)$

$KeyData = kdf(Z_s)$

# ANSI X9.63 - 1-Pass UM



$Q_{s,U}$



$Q_{s,V}$

$V$

$U$



$Q_{e,U}$



$$Z_s = [h]d_{s,U}Q_{s,V}$$

$$Z_e = [h]d_{e,U}Q_{s,V}$$

$$KeyData = kdf(Z_e || Z_s)$$

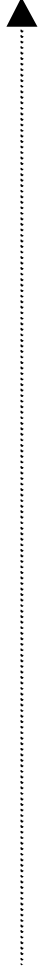
$$Z_s = [h]d_{s,V}Q_{s,U}$$

$$Z_e = [h]d_{s,V}Q_{e,U}$$

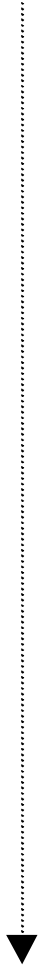
$$KeyData = kdf(Z_e || Z_s)$$

# ANSI X9.63 - Full UMI

$Q_{s,U}$



$Q_{s,V}$



$U$

$Q_{e,U}$



$Q_{e,V}$



$Z_s = [h]d_{s,U}Q_{s,V}$

$Z_e = [h]d_{e,U}Q_{e,V}$

$KeyData = kdf(Z_e || Z_s)$

$Z_s = [h]d_{s,V}Q_{s,U}$

$Z_e = [h]d_{e,V}Q_{e,U}$

$KeyData = kdf(Z_e || Z_s)$

$V$



# ANSI X9.63 - 1-Pass MQV



$Q_{s,U}$



$Q_{s,V}$

$U$



$V$

$Q_{e,U}$



$$\text{implicitSig}_U = d_{e,U} + \text{avf}(Q_{e,U}) d_{s,U}$$

$$R = h \times \text{implicitSig}_U \times (Q_{s,V} + \text{avf}(Q_{s,V}) Q_{s,V})$$

$$Z = x_R$$

$$\text{KeyData} = \text{kdf}(Z)$$

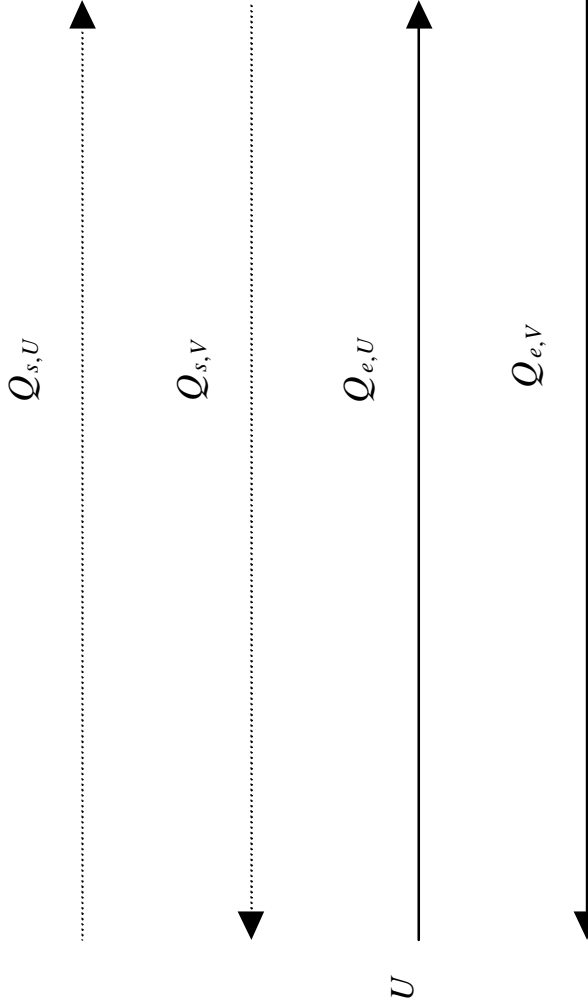
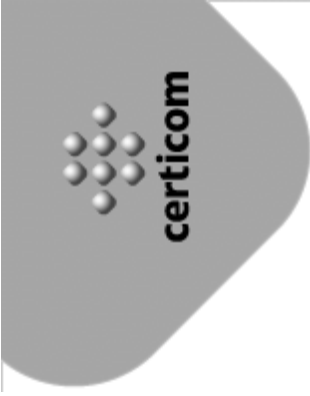
$$\text{implicitSig}_V = d_{s,V} + \text{avf}(Q_{s,V}) d_{s,V}$$

$$R = h \times \text{implicitSig}_V \times (Q_{e,U} + \text{avf}(Q_{e,U}) Q_{s,U})$$

$$Z = x_R$$

$$\text{KeyData} = \text{kdf}(Z)$$

# ANSI X9.63 - Full MQV



$$\text{implicit}_{sig_U} = d_{e,U} + \text{avf}(Q_{e,U}) d_{s,U}$$

$$R = h \times \text{implicit}_{sig_U} \times (Q_{e,V} + \text{avf}(Q_{e,V}) Q_{s,V})$$

$$Z = x_R$$

$$\text{KeyData} = \text{kdf}(Z)$$

$$\text{implicit}_{sig_V} = d_{e,V} + \text{avf}(Q_{e,V}) d_{s,V}$$

$$R = h \times \text{implicit}_{sig_V} \times (Q_{e,U} + \text{avf}(Q_{e,U}) Q_{s,U})$$

$$Z = x_R$$

$$\text{KeyData} = \text{kdf}(Z)$$

# ANSI X9.63 - 1-Pass Transport



$Q_{enc,V}$



$U$

$V$

$ENC_{Q_{enc,V}}(U \parallel KeyData)$



$KeyData = KeyData$

$U \parallel KeyData =$

$DEC_{d_{enc,V}}(ENC_{Q_{enc,V}}(U \parallel KeyData))$

$KeyData = KeyData$

## **ANSI X9.63 - Additional Schemes**

ANSI X9.63 also specifies a variety of schemes that provide explicit key authentication and entity authentication:

- Combined UM with key confirmation
- Full UM with key confirmation
- Station-to-Station
- Full MQV with key confirmation
- 3-pass key transport

(Explicit key authentication and entity authentication are often essential in applications.)

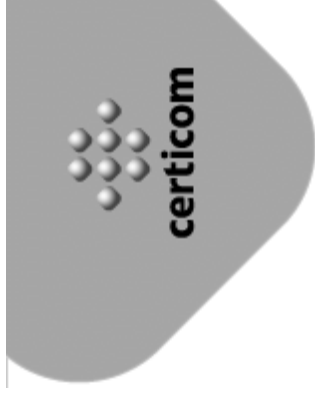


## **Does X9.63 meet NIST's needs?**

ANSI X9.63 is designed to meet the financial services industry's needs but also appears suitable for government needs.

Elliptic curve cryptography offers extra efficiency suitable for securing advanced applications like e-commerce and smart card based security

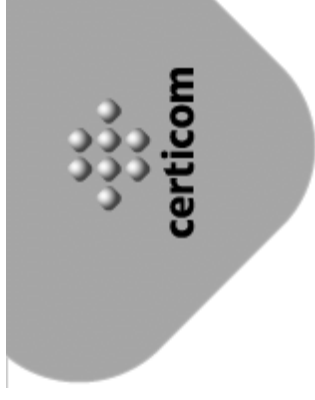
Standard relies on NIST approved building blocks like SHA-1



## **Does X9.63 meet NIST's needs? (2)**

Patent call has been issued and patent letters have been received from Certicom and IBM

Standard aligned with ANSI X9.42, FIPS 196, IEEE P1363, ISO 9798-3, ISO 11770-3, ISO 15946-3



## Conclusion

- ANSI X9.42, X9.44, and X9.63 appear to provide a suitable basis for a key establishment FIPS
- Variety of schemes allows future application developers to meet the needs of a diverse range of applications
- Choice of schemes will be limited by application requirements
- This policy could lead to development of “FIPS compliant versions of existing standards like IPsec, S/MIME, TLS
- Future standards could be developed with FIPS in mind.

