PUBLIC-KEY BASED KEY-ESTABLISHMENT SP 800-56 SERIES

Lily Chen

Computer Security Division

Information Technology Lab, NIST

Some questions

- Why public-key based key establishment schemes were not specified in FIPS documents but SPs? (Digital signature schemes were specified in FIPS 186)
- How the decisions were made on which schemes to be included in SP 800-56A and SP 800-56B
- How the schemes specified in SP 800-56A and SP 800-56B are used in security protocols such as TLS and IKE
- Whether PQC ML-KEM specified in FIPS 203 be a “drop-in-replacement” for any scheme in SP 800-56A or SP 800-56B

Outline

- Background and history of 56A and 56B
- Schemes included
- Key derivation methods (56C)
- Public-key based key establishment in IETF
- What we learned

General background on Public-Key Based Establishment

- Two major categories
  - Discrete logarithm based Diffie-Hellman key agreement (or key exchange) – proposed in the paper "New Direction in Cryptography" published in 1976
  - Factorization based RSA public key encryption based on the paper "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems" published in 1978
- Early external standardization effort
  - PKCS #1 by RSA Labs (1991 -)
  - X9 (1994-)
  - IEEE P1363 (1994-)
  - Standards for Efficient Cryptography Group (SECG) on ECC (1998-)
  - IETF (IKE v1, 1998 and SSL 2.0, 1995 then TLS 1.0, 1999)

Note: FIPS 186 Digital Signature Standard was published in 1994

SP 800-56A (Background and History)

- SP 800-56A Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography
- Schemes specified over finite fields and elliptic curves based on
  - Diffie-Hellman
  - MQV
- Adopted from X9.42 (over finite fields) and X9.63 (over elliptic curves) – standards developed by X9 for the financial industry
- Revision history
  - The first version March 2006 (the development started in 2001)
  - SP 800-56A rev1 March 2007 (minor changes)
  - SP 800-56A rev2 May 2013 (revision started in 2010)
  - SP 800-56A rev3 April 2018

SP 800-56A primitive 1 – Diffie-Hellman (over a finite field)

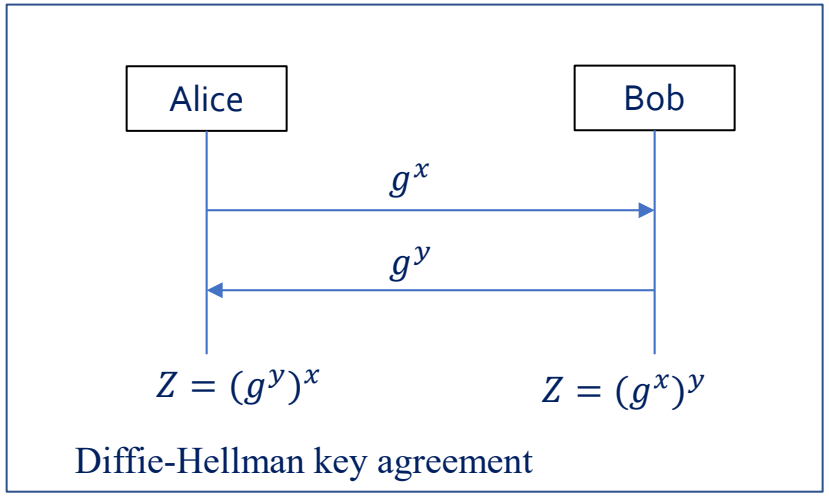
- Diffie-Hellman over finite field $\mathbb{F}_p$ where p is a prime.
  - The scheme is defined over multiplicative cyclic subgroup $\mathbb{G}$ with order q.
  - g is a generator of $\mathbb{G}$, that is, $\mathbb{G} = \{1, g, g^2, \dots, g^{q-1}\}$

1. Alice randomly selects an integer $x \in \{0, 1, \dots, q-1\}$, computes $Y_a = g^x \ mod \ p$ and sends Y_a to Bob.
   - x is called Alice's private key and Y_a Alice's public key.

2. Bob randomly selects an integer $y \in \{0, 1, \dots, q-1\}$, computes $Y_b = g^y \ mod \ p$ and sends Y_b to Alice.
   - y is called Bob's private key and Y_b Alice's public key.

3. Upon receiving Y_b from Bob, Alice computes
   $$Z = (Y_b)^x = (g^y)^x = g^{yx} \ mod \ p.$$

4. Upon receiving Y_a from Alice, Bob computes
   $$Z = (Y_a)^y = (g^x)^y = g^{xy} \ mod \ p.$$

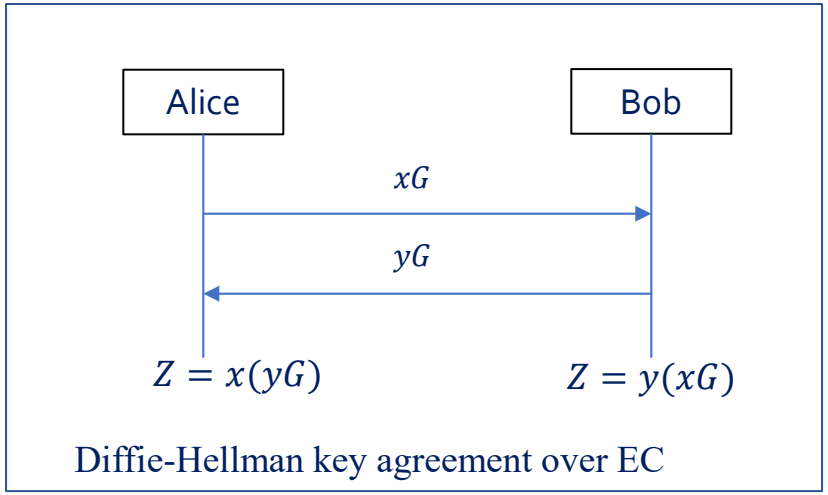


Diffie-Hellman key agreement

SP 800-56A primitive 1 – Diffie-Hellman (over an elliptic curve)

- Diffie-Hellman over an elliptic curve $\mathbb{E}(\mathbb{F}_q)$ where q is a prime or $\mathrm{q} = 2^m$, where m is an integer.
  - The scheme is defined over a subgroup $\mathbb{G}$ of $\mathbb{E}(\mathbb{F}_q)$ with order n.
  - G is a generator of $\mathbb{G}$, that is, $\mathbb{G} = \{\phi, G, 2G, \dots, (n-1)G\}$, where for an integer d, $dG = G + G + \cdots + G$, conducting operation "+" $d - 1$ times over elliptic curve and ϕ is infinite point $\phi = nG$.

1. Alice randomly selects an integer $x \in \{0, 1, \dots, n-1\}$ and computes $Y_a = xG$ and sends Y_a to Bob.
2. Bob randomly selects an integer $y \in \{0, 1, \dots, n-1\}$ and computes $Y_b = yG$ and sends Y_b to Alice.
3. Upon receiving Y_b from Bob, Alice computes
$$\mathrm{Z} = \mathrm{x}(Y_b) = x(yG) = (xy)G.$$
4. Upon receiving Y_a from Alice, Bob computes
$$\mathrm{Z} = \mathrm{y}(Y_a) = y(xG) = (yx)G.$$



Diffie-Hellman key agreement over EC

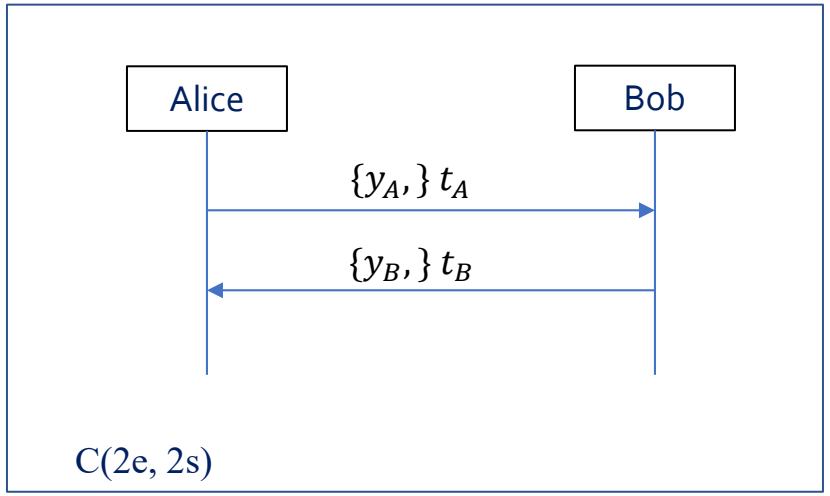
SP 800-56A -Diffie-Hellman schemes

- For a Diffie-Hellman key agreement, the key can be ephemeral or static
- Each party can use one pair of keys or two pairs of keys
  - C(2e, 2s): each party uses two pairs of keys, one pair static while the other ephemeral; it generates two "shared secret" values Z_1 and Z_2.
  - C(2e, 0s): each party uses one pair of ephemeral keys; it generate one "shared secret" value Z
  - C(1e, 2s): one party uses only one pair of static keys, another party uses one pair of ephemeral keys and one pair of static keys; it generates two "shared secret" values Z_1 and Z_2.
  - C(1e, 1s): one party uses one pair of static keys, while another party uses one pair of ephemeral keys; ; it generate one "shared secret" value Z
  - C(0e, 2s): each party uses one pair of static keys; it generates one shared secret value Z.
- The operations can be over a finite field or an elliptic curve
  - In case of two pairs of keys are used for one party, both key pairs must be on the same finite field or both on the same elliptic curve (56A specific)

SP 800-56A – DH schemes with two pair of keys for each party

- Each party has one pair of static keys and one pair of ephemeral keys
  - Alice: (x_A, y_A) – static and (r_A, t_A) – ephemeral
  - Bob: (x_B, y_B) – static and (r_B, t_B) – ephemeral
- Alice's operations

1. $Z_1 = {y_B}^{x_A}$
2. $Z_2 = {t_B}^{r_A}$
3. $Z = Z_1 \parallel Z_2$



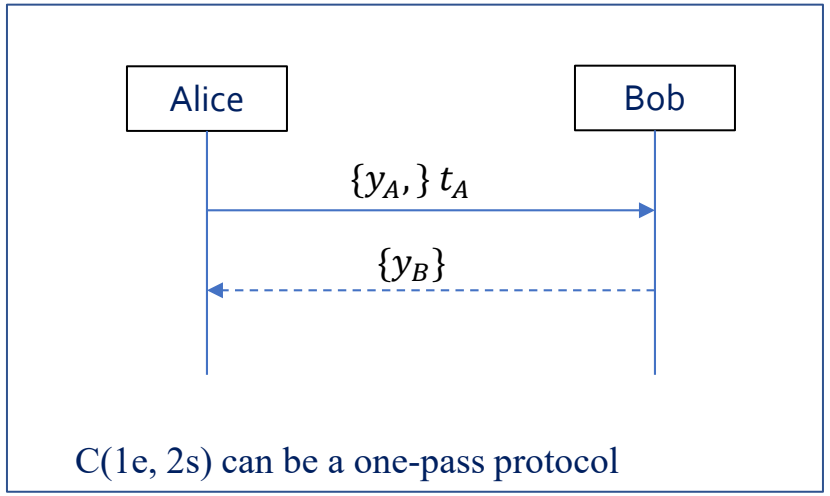
SP 800-56A – DH schemes with two pair of keys for one party

- Alice uses two pairs of keys, while Bob uses only static key pair
  - Alice: (x_A, y_A) – static and (r_A, t_A) – ephemeral
  - Bob: (x_B, y_B) – static
- Alice's operations

1. $Z_1 = {y_B}^{x_A}$
2. $Z_2 = {y_B}^{r_A}$
3. $Z = Z_1 \parallel Z_2$

- Bob's operation

1. $Z_1 = {y_A}^{x_B}$
2. $Z_2 = {t_A}^{x_B}$
3. $Z = Z_1 \parallel Z_2$



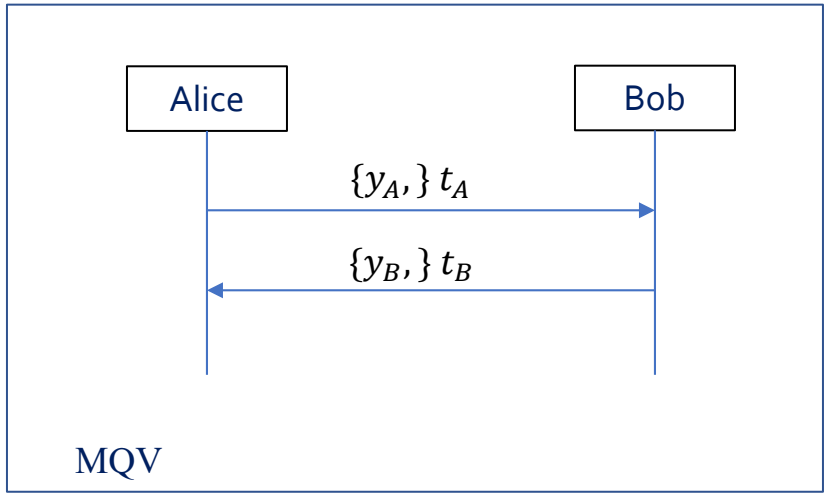
Components of key establishment

- Domain parameters
  - finite field: (p, q), generator g (optionally seed and counter)
  - elliptic curve: (a, b), $y^2 = x^3 + ax + b$ (q is a prime) or $y^2 + xy = x^3 + ax^2 + b$ ($q = 2^m$ and field representation), generator G, n (the order of G), cofactor h, that is, the number of points on the elliptic curve is nh.
- Key pair generation (ref. random number generation SP 800-90 series, FIPS 186, …)
- Kay validation (key pair for owner, public-key for recipient, …)
- Key derivation: Derive keying material from shared secret value Z
- Key confirmation (bilateral or unilateral): a party confirms to another party it has obtained the keying material, as a result of execution of key establishment
  - When static key is used, if the static public-key is certified, then key confirmation is also an implicit entity authentication

SP 800-56A primitive 2 – MQV (over a finite field)

- MQV was initially proposed by Alfred Menezes, Minghua Qu and Scott Vanstone in 1995.
- MQV over finite field $\mathbb{F}_p$ where p is a prime.
  - The scheme is defined over multiplicative cyclic subgroup $\mathbb{G}$ with order q.
  - g is a generator of $\mathbb{G}$, that is, $\mathbb{G} = \{1, g, g^2, \dots, g^{q-1}\}$
- Each party has one pair of static keys and one pair of ephemeral keys
  - Alice: (x_A, y_A) – static and (r_A, t_A) – ephemeral
  - Bob: (x_B, y_B) – static and (r_B, t_B) – ephemeral
- Alice's operations

1. $w = \left\lceil \frac{1}{2} log_2 q \right\rceil$
2. $T_A = (t_A mod\ 2^w) + 2^w$
3. $S_A = (r_A + T_A x_A)\ mod\ q$
4. $T_B = (t_B\ mod\ 2^w) + 2^w$
5. $z = \left((t_B ({y_B}^{T_B})^{S_A})\right) mod\ p = (g^{r_B + x_B T_B})^{r_A + x_A T_A}\ mod\ p$



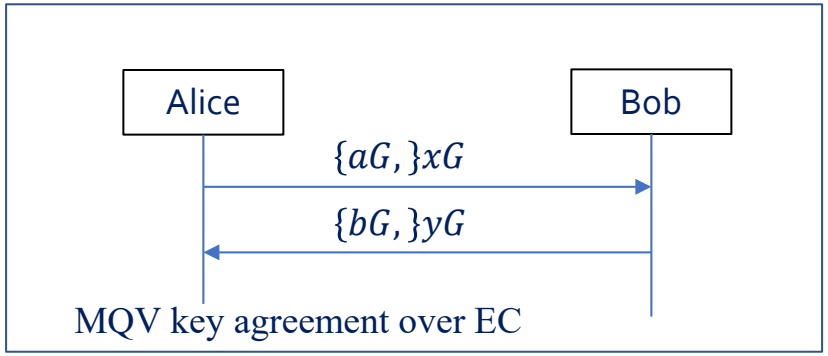
SP 800-56A primitive 2 – MQV (over an elliptic curve)

- MQV over an elliptic curve $\mathbb{E}(\mathbb{F}_q)$where q is a prime or q $= 2^m$, where m is an integer.
  - The scheme is defined over a subgroup $\mathbb{G}$ of $\mathbb{E}(\mathbb{F}_q)$ with order n. The number of points on $\mathbb{E}$ is nh, where h is the cofactor.
  - G is a generator of $\mathbb{G}$, that is, $\mathbb{G} = \{\phi, G, 2G, \dots, (n-1)G\}$, where for an integer d, $dG = G + G + \cdots + G$, conducting operation "+" $d - 1$ times over elliptic curve and ϕ is infinite point $\phi = nG$.
- Each party has two pair of keys(simplified the notations)
  - Alice: (a, A) – static and (x, X) – ephemeral, where $A = aG$ and $X = xG$
  - Bob: (b, B) – static and (y, Y) – ephemeral, where $B = bG$ and $Y = yG$
- Alice's operations

1. Denoting $X = (u_X, v_X)$ and $Y = (u_Y, v_Y)$, compute

$\bar{u}_X = (u_X \ mod \ 2^L) + 2^L$ and $\bar{u}_Y = (u_Y \ mod \ 2^L) + 2^L$, where $L = \left\lceil \frac{log_2 n}{2} \right\rceil$

2. $I_A = (x + \bar{u}_X a) \ mod \ n \ \ and \ \ P = hI_A(Y + \bar{u}_Y B)$
3. Denoting $P = (u_P, v_P), z = u_P$.



SP 800-56A -MQV schemes

- For a MQV key agreement scheme, there are two options
  - C(2e, 2s): (MQV2 and full MQV) each party uses two pairs of keys, one pair static while the other ephemeral as in the primitive
  - C(1e, 2s): (MQV 1 and one-pass MQV) one party uses only one pair of static keys, another party uses one pair of ephemeral keys and one pair of static keys
    - In the MQV operation, the party with only one pair of static keys will use the same key pair in the place of ephemeral key
    - The main purpose is to make a one-pass protocol, assuming the static public-key already shared
- The operations can be over a finite field or an elliptic curve

SP 800-56A – Research on MQV schemes

- MQV was initially proposed by Alfred Menezes, Minghua Qu and Scott Vanstone in 1995 (late, a more formal paper published in 2001)
  - explicit key confirmation is optional
- In 2001, Kaliski presented an unknown key-share attack that exploited the missing identities in the MQV key exchange protocol
  - In 2006, Menezes and Ustaoglu proposed to address this attack by including user identities in the key derivation function at the end of the MQV key exchange
- In 2005, Krawczyk proposed a hash variant of MQV, called HMQV, on EC version of MQV
  - hash x component with identity for each party's ephemeral key
  - Remove check the proof-of-possession of the user's static private key by CA and ephemeral public key validation by recipient to save cost
- In 2006, in response to Menezes's attack, Krawczyk revised HMQV
  - validate them together in one combined operation- the version is included in IEEE P1363
- In 2010, Hao presented two attacks on the revised IEEE P1363 version of HMQV

More on schemes in SP 800-56A

- SP 800-56A specified 14 schemes (Table 8). Each of them is determined by
  - Key pairs used for each party
  - Over finite field or elliptic curve
  - MQV or DH in C(2e, 2s) and C(1e, 2s)
- Cofactor ECC DH
  - Z is computed by Alice with key pair (x, Y_A) and Bob's public key Y_B as

    $Z = xhY_B = xyhG$, where h is the cofactor.

    If $Z = \phi$, then return error.
  - The cofactor ECC DH is to save checking $nY_B \neq \phi$ in full public-key validation, since nY_B is considered as an "expensive" computation.

SP 800-56A: Major revisions and related issues

- SP 800-56A (2006) – SP 800-56A rev1 (2007): make an exception for static key pair usage for signature to confirm possession of the private key
  - **Note**: To use signature to confirm private key possession, in finite field, $p = hq$, that is, p and q must satisfy DSA requirements on parameters, for example, when $|p| = 2048,\ |q| = 224$ or 256
- SP 800-56A rev1 (2007) – SP 800-56A rev2 (2013): many changes, key derivation refers 56C for two-step key derivation, and SP 800-135 for application specific KDFs
- SP 800-56A rev 2(2013) – SP 800-56A rev3 (2018):
  - approve finite field defined by safe-prime ($p = 2q$) defined in RFC 3526 (IKE) and RFC 7919 (TLS). "FIPS 186-type domain parameters should only be used for backward compatibility with existing applications that cannot be upgraded to use the approved safe-prime groups."
  - approve elliptic curves defined in RFC 4492 (TLS) and in RFC 5903 (IKEv2), in addition to curves specified in FIPS 186-4 (SP 800-186 was under development at the time)
  - Move all key derivations to SP 800-56C

SP 800-56B (Background and History)

- SP 800-56A Recommendation for Pair-Wise Key-Establishment Schemes Using Integer Factorization Cryptography
- Schemes specified are based on RSA encryption primitive
- Adopted from X9.44 – standards developed by X9 for the financial industry
- Revision history
  - The first version August 2009 (development began in 2006)
  - SP 800-56B rev1 September 2014 (revision began in 2012)
  - SP 800-56A rev2 March 2019

SP 800-56B RSA key pairs, formats, and generation

- RSA key pair
  - Public key: (n, e)
  - Private key: (n, d)

  where $n = pq$, both p and q are prime and $ed = 1 \; mod \; \phi(n)$.

- Formats of private key
  - Basic format: (n, d)
  - Prime-factor format: (p, q, d)
  - Chinese remainder theory (CRT) format:

    $$(n, e, d, p, q, dP, dQ, q^{-1} mod \; p),$$

    where $dP = d \; mod \; (p - 1)$, $dQ = d \; mod \; (q - 1)$.

- The RSA prime generation is specified in FIPS 186
  - p and q must satisfy certain requirements
  - Public component e is an odd number, satisfying $65{,}5537 \leq e < 2^{256}$ and e is relatively prime to $p - 1$ and $q - 1$
    - e can be fixed for all selections of p and q or randomly selected for each selection of p and q
  - Private component $d = e^{-1} \; mod \; \phi(n)$, that is, $ed = 1 \; mod \; \phi(n)$. If $d \leq 2^{\frac{|n|}{2}}$, discard and start over.

SP 800-56B primitive – RSA encryption and decryption

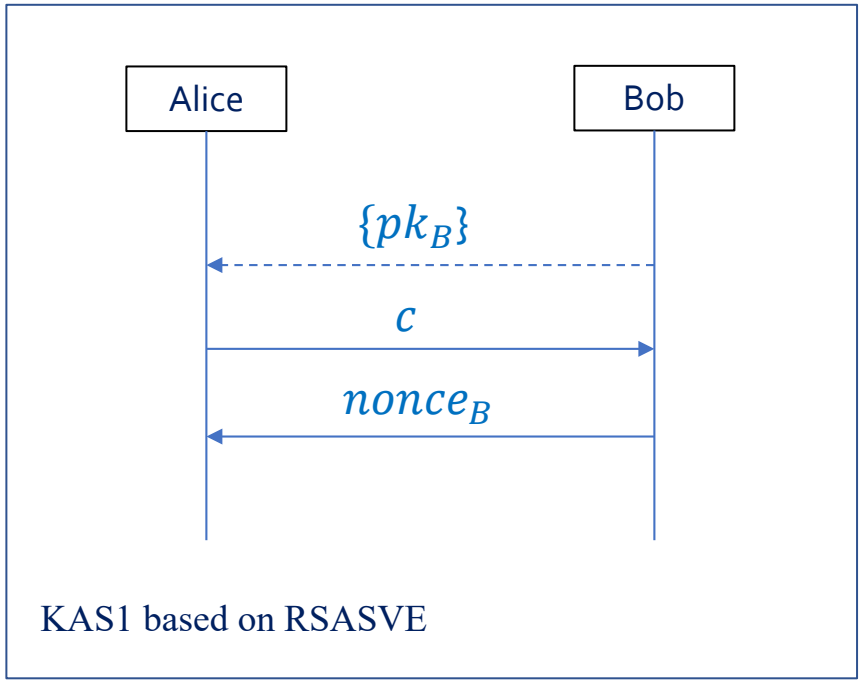
- Encryption
  - Public key (n, e) and integer $m, 1 < m < n - 1, c = m^e \ mod \ n$
- Decryption
  1. With basic format private key $(n, d), m = c^d \ mod \ n$
  2. With prime factor format private key $(p, q, d), n = pq, m = c^d \ mod \ n$
  3. With CRT format private key $(n, e, d, p, q, dP, dQ, q^{-1} mod \ p)$, conduct the following operations.
     a. $m_p = c^{dP} \ mod \ p$
     b. $m_q = c^{dQ} mod \ q$
     c. $h = \left((m_p - m_q) q^{-1}\right) mod \ p$
     d. $m = ((m_q + (q \times h)) \ mod \ n$

SP 800-56B – RSA Secret-Value Encapsulation (RSASVE)

- RSASVE Generate Operation
  - Public key (n, e)
  - Randomly generate an integer z, $1 < z < n - 1, c = z^e \ mod \ n$
  - Output c and z
- RSASVE Recovery Operation
  - RSA private key and ciphertext $((n, d),\ c),\ z = c^d \ mod \ n$
  - Output z

SP 800-56B – Key agreement schemes based on RSASVE

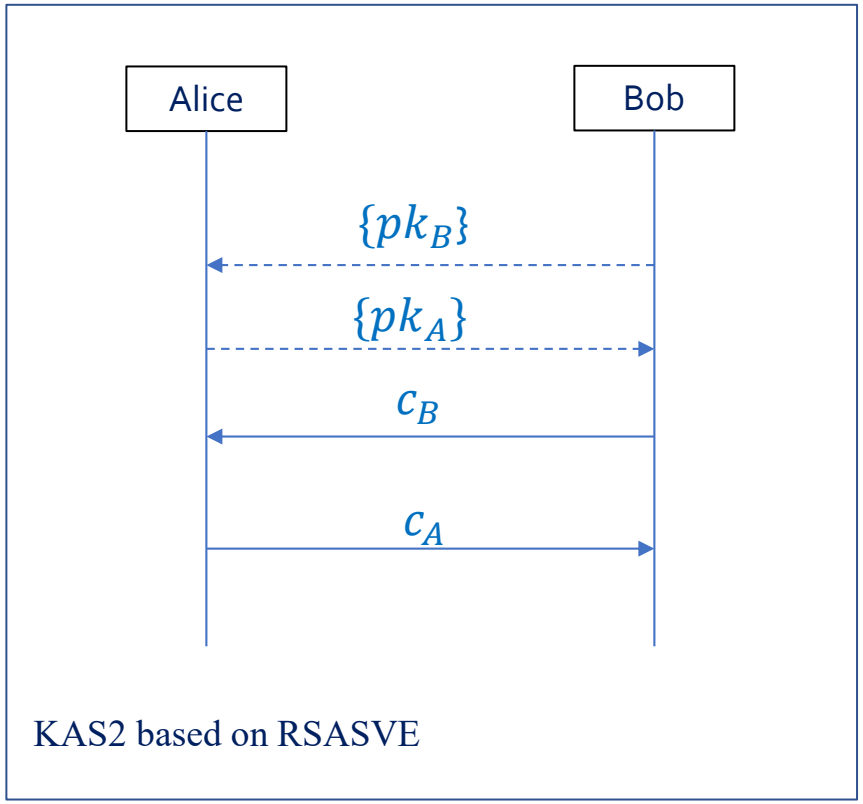
- KAS1
  - Alice uses Bob's public key and RSASVE generate operation to generate a cipher text c and secret value z and sends c to Bob
  - Bob uses RSASVE recovery operation and his private key to obtain z, randomly selects a nonce $nonce_B$, and sends to Alice
  - The keying material is derived from z as shared secret and having $nonce_B$ included in the "$OtherInput$".
  - Optionally, Bob sends key confirmation to Alice



KAS1 based on RSASVE

SP 800-56B – Key agreement schemes based on RSASVE

- KAS2
  - Alice and Bob separately conduct RSASVE generation operation to generate (z_A, c_A) and (z_B, c_B), respectively
  - Alice and Bob exchange ciphertext c_A and c_B, then conducts RSASVE discovery operation to obtain z_A and z_B
  - The keying material is derived with $z_A \parallel z_B$ as shared secret
  - Optionally, Alice and/or Bob provide key confirmation



KAS2 based on RSASVE

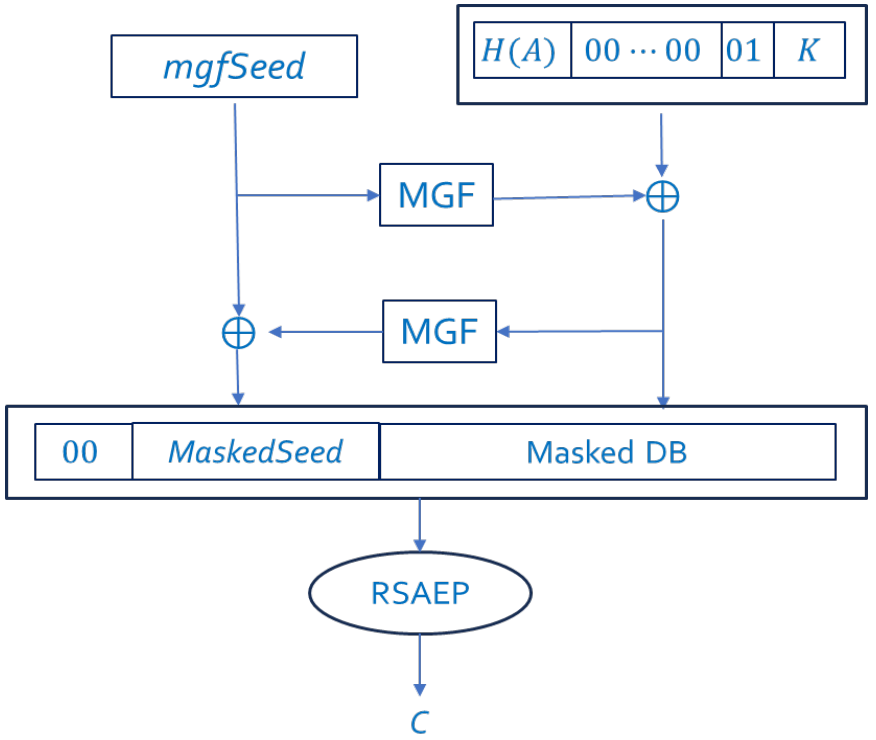
SP 800-56B – RSA-OAEP

- OAEP stands for Optimal Asymmetric Encryption Padding
- It was introduced by Bellare and Rogaway in 1994 with an **incorrect** security proof of chosen ciphertext secure for any trapdoor permutation in random oracle model
  - In 2000, Okamoto and Stern proved that RSA-OAEP is indeed chosen ciphertext secure by using special property of RSA
- RSA-OAEP is a probabilistic encryption by calling random bits (seed) and using a mask generation function (MGF)

SP 800-56B – RSA-OAEP encryption

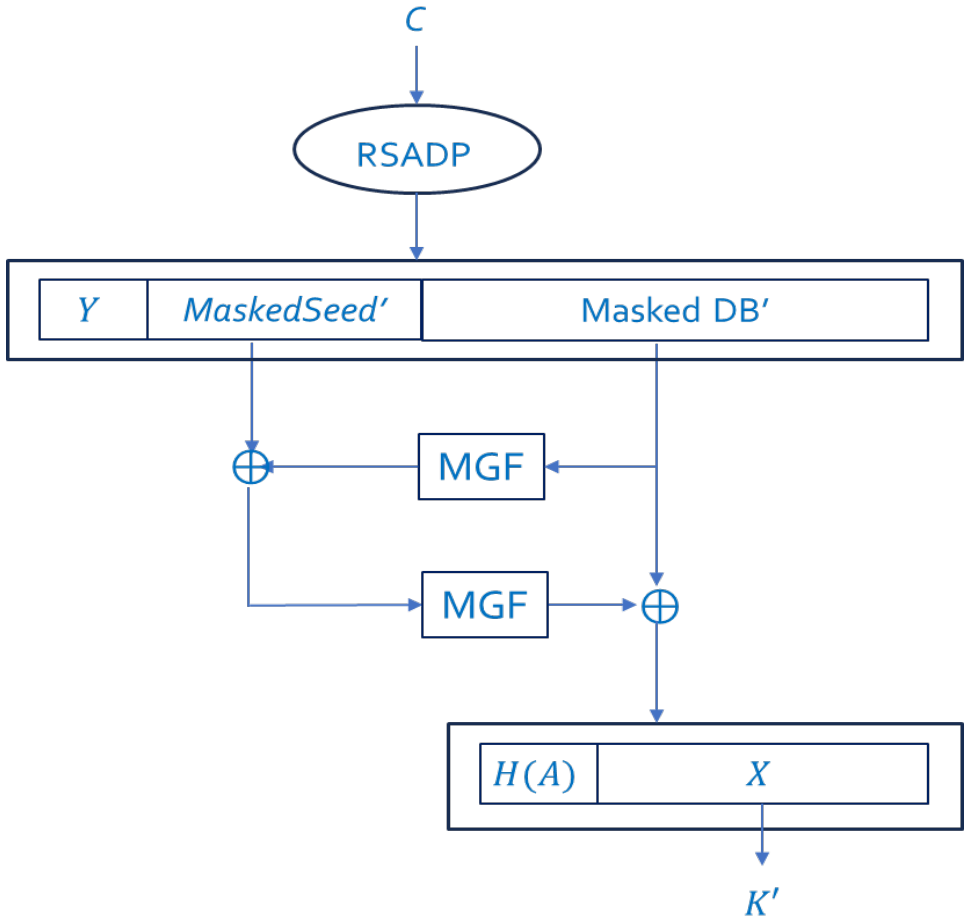
- RSA-OAEP encryption
  - Input: (n, e) – public key; K – keying material to be encrypted; A- additional input
  - Output: C – ciphertext
- Auxiliary functions
  - H – Hash function
  - MGF – Mask generation function
- Notations of data lengths in bytes
  - nLen – length of integer n
  - HLen – length of hash H output
  - Klen – Length of K
- It must satisfy

$$nLen - 2HLen - KLen - 2 \geq 0$$



SP 800-56B – RSA-OAEP decryption

- RSA-OAEP decryption
  - Input: C – Ciphertext, (n, d) – RSA private key
  - Output: K – key
- Auxiliary functions
  - H – Hash function
  - MGF – Mask generation function
- Notations of data lengths in bytes
  - nLen – length of integer n
  - HLen – length of hash H output
  - Klen – Length of K



SP 800-56B – Key transport schemes

- SP 800-56B specifies two key transport schemes

1. KTS-OAEP: OAEP based key transport scheme with option of key confirmation with the recipient as the provider
2. Hybrid Key-Transport Methods: Following KAS1, KAS2 or KTS-OAEP with a key wrapping (SP 800-38F)

SP 800-56B: Major revisions

- SP 800-56B(2009) – SP 800-56B rev1 (2014): changes in presentations, introduced two-step key derivation, and removed 80-bits security parameter
- SP 800-56B rev1 (2014) - SP 800-56B rev2 (2019): Removed reference to TDES, removed KTS-KEM-KWS scheme and added KTS-Hybrid-SKW, Provided moduli > 3072 bits, added reference of key derivation to 56C

  - KTS-KEM-KWS.encryption of a key K is a scheme with the following major steps:

  1. Use RSASVE to generate (C_1, Z)
  2. Derive a key-warping-key from Z
  3. Use the key-wrapping-key to wrap K (SP 800-38F) to obtain ciphertext C_2
  4. The ciphertext $C = C_1 \parallel C_2$

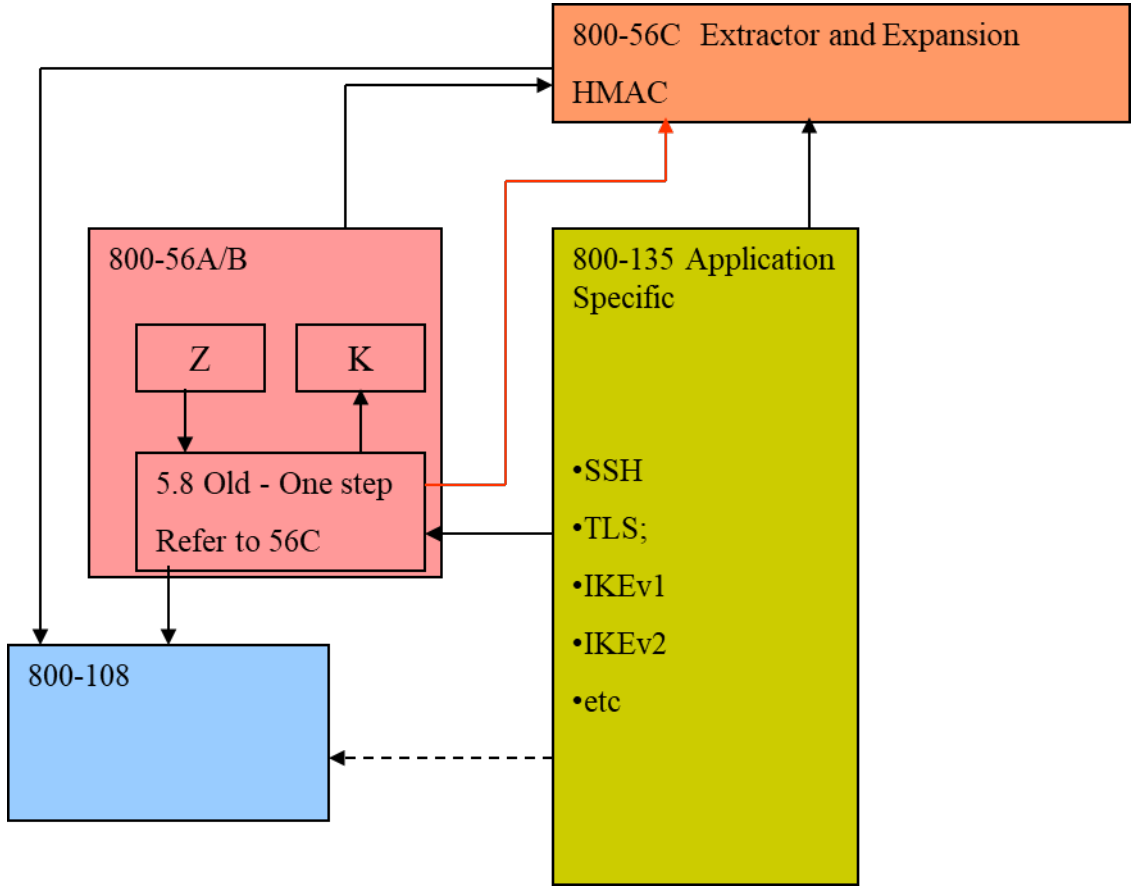
SP 800-56C (Background and History)

- SP 800-56A (2006), SP 800-56A rev1 (2007), and SP 800-56B (2009) specified one-step key derivation using hash function with input formats
  - Concatenation
    - $H_i = H(i \parallel Z \parallel OtherInfo)$,

      where $Otherinfo = AlgorithmID \,||\, PartyUInfo \,||\, PartyVInfo \,\{||\, SuppPubInfo\,\}\{||\, SuppPrivInfo\,\}$
    - $K = H_1 \parallel H_2 \parallel \cdots \parallel H_n$ and Derived keying material is the first L bits of K
  - ASN.1.
    - It is essentially the same as concatenate key derivation but utilizing ASN.1 DER encoding of $OtherInfo$.
- Hugo Krawczyk 2008 paper "On Extract-then-Expand Key Derivation Functions and an HMAC-based KDF" commented NIST KDFs defined in SP 800-56A (and SP 800-56B) as "ad hoc" manner and proposed to explicitly conduct extraction before expansion.
- In 2009, when SP 800-56A was under review, two-step key derivation was considered
- Revision history
  - The development of SP 800-56C started in 2010 and published in 2011
  - SP 800-56C rev1 April 2018
  - SP 800-56C rev2 August 2020

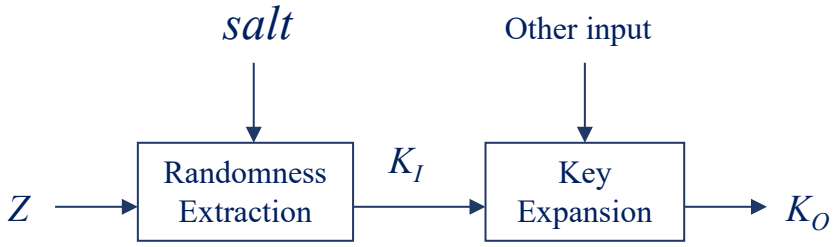
SP 800-56C: complicated decisions to make

- Key derivation had been specified by many standards, besides 56A/B, e.g.
  - X9F1 specifies KDFs in X9.42, X9.63, X9.44
  - IEEE P1363
  - ISO/IEC 11770-6
  - IETF (TLS, IKE, …)
  - IEEE 802 wireless standards
- Most of the designs were "ad hoc" and it has been introduced many variants
- It has been hard to cover all the situations
  - Received a lot of questions on key derivation for CMVP validation



SP 800-56C: Two-step Key Derivation

- Extraction
  - $K_{DK} = MAC(salt, Z, \cdots)$
- Expansion
  - $DerivedKeyingMaterial = KDF(K_{DK}, L, \{IV\}, FixedInfo)$, where KDF is one the key derivation functions specified in SP 800-108



SP 800-56C: Revisions

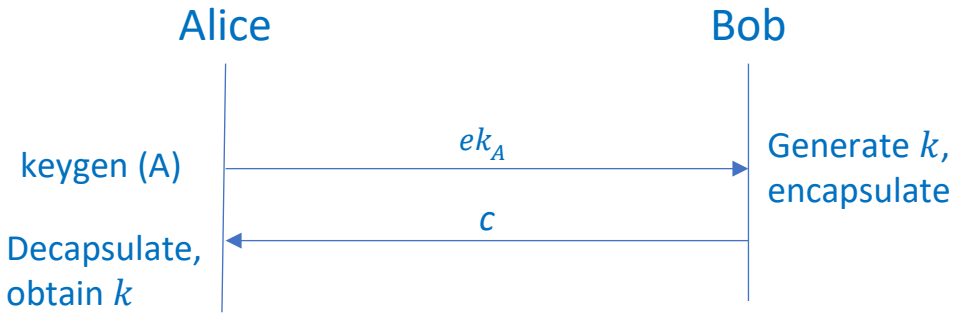
- SP 800-56C (2011) – SP 800-56C Rev1 (2018):
  - Moved detailed one-step key derivation to SP 800-56C from 56A/B, added KMAC based key derivation for one-step auxiliary function (but not for two steps)
- SP 800-56C rev1 (2018) – SP 800-56C rev2 (2020):
  - Explicitly allow to enter additional "shared secret" to accommodate hybrid mode, that is, the shared secret can be $Z \parallel T$, where Z is obtained by an approved scheme in 56A or 56B and T can be obtained by any scheme(s)
  - Allow to make multiple expansions with different labels after extraction step to accommodate TLS 1.3 key derivation

Key establishment in TLS and IKE

- For public-key based Internet Key Exchange (IKEv1 and IKEv2), it uses ephemeral-ephemeral key Diffie-Hellman key agreement over pre-defined finite fields and elliptic curves
- For TLS 1.2 and earlier version(s), the key exchange cipher suites support
  - Ephemeral-ephemeral (DHE)
  - Ephemeral for client and static for server (DH)
  - RSA with client generated pre-master secret and encrypted with server's public-key
- TLS 1.3 only supports DHE over pre-defined finite fields or elliptic curves

Migration to post-quantum key establishment

- ML-KEM specified in FIPS 203 can be implemented in the place of DH and RSA for security protocols with advanced security proofs
  - One party contributes encapsulation key while another party generates the key and encapsulates it
  - Many different variants can be used in protocols
- Hybrid key establishment uses key derivation as a combiner for the secret values/keys obtained from different schemes
  - Some introduced a term "combiner"



- Draft SP 800-227 "Recommendations for Key-Encapsulation Mechanisms" was released for public comments (due March 7, 2025)
- Workshop on PQ KEM, Feb. 25-26, 2025 (virtual)

What we learned

- Ideas or proposals from researchers, even with some advantages, may not be adopted by applications, e.g.
  - C(2e, 2s) and C(1e, 2s) schemes including MQV

  Selection of schemes must consider the adoption interests from application community

- Some security features can only be achieved through a properly designed protocol e.g.
  - implicit authentication using certified static key with key confirmation

  It must be clear about how to achieve the expected security features

- Some security requirements may get practical difficulties, e.g.
  - Full public-key validation (costly in ECDH, not possible for RSA)
  - Assurance of private-key possession for the public-key recipient
  - Include identifiers for both parties in key derivation

  Identify requirements with specific schemes and security concerns

Summary

- Public-key cryptography enables establishing symmetric keys for data protection through public channel
- SP 800-56A and SP 800-56B were based on standards developed in X9, effort began in 1994
  - 56A and 56B included additional requirements to enable CAVP validation
- Each SP in the series have experienced multiple revisions
  - Major changes were triggered by research results or to synch with other standards
- Currently, SP 800-56 series are under review in "Crypto Publication Review Project" - The public comment period was closed on January 31, 2025
  - Thank you for the comments!