Topic 3.5

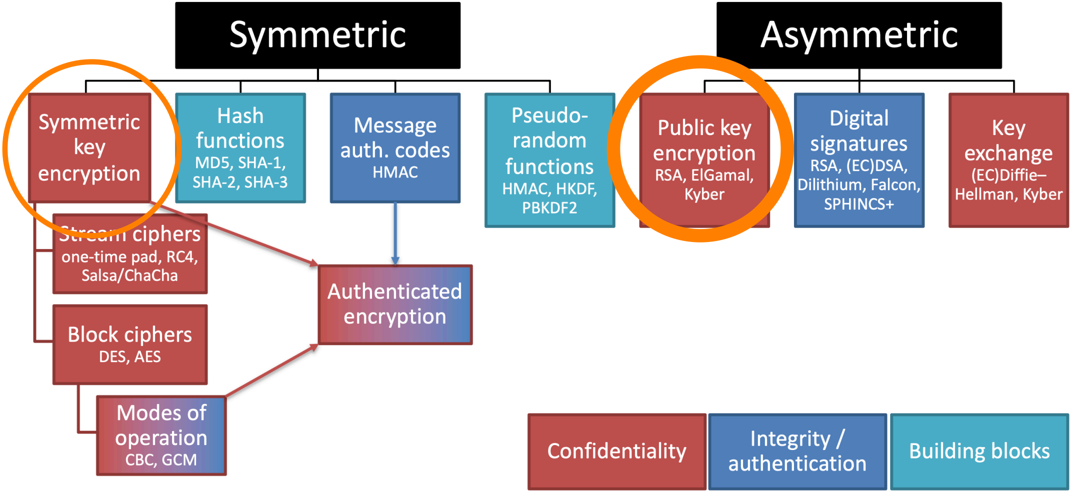
Public key cryptography – Hybrid encryption

Douglas Stebila

CO 487/687: Applied Cryptography

Fall 2024

Map of cryptographic primitives



Symmetric-key vs. public-key

Symmetric-key encryption:

- Fast!
- Any bitstring of the right length is a valid key.
- Any bitstring of the right length is a valid plaintext.
  - Stream ciphers have no length restrictions on the plaintext.
  - Block ciphers have fixed-length plaintexts but support modes of operation (e.g. CBC) with arbitrary message lengths.
- Security assumptions are based on published analyses and attempted attacks, but are not directly linked to “natural” mathematical problems.
- Typical attack speed: $\approx 2^\ell$ operations where ℓ is the key length.

Symmetric-key vs. public-key

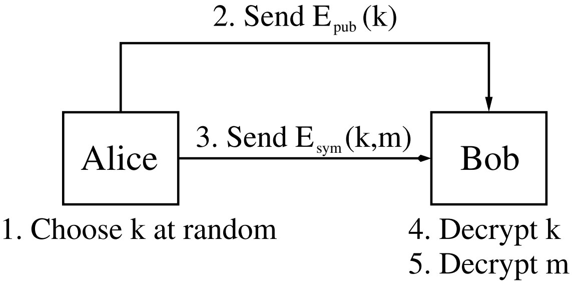
Public-key encryption:

- Slow!
- Keys have special structure—not every bitstring of the right length is a valid key.
- Not every bitstring of the right length is a valid plaintext. Typical message spaces include:
  - (RSA) $M = \mathbb{Z}_n^* = \{x \in \mathbb{Z}_n : \gcd(x, n) = 1\}$
  - (Elgamal) $M = \mathbb{Z}_p^* = \{1, 2, \ldots, p-1\}$
- Security assumptions are provably linked to "natural" mathematical problems such as factoring.
- Typical attack speed: Much faster than $\approx 2^\ell$ operations! (where ℓ is the key length).

Hybrid encryption

Basic idea:

1. Use public-key encryption to establish a shared secret key
2. Use symmetric-key encryption with the shared secret key to encrypt data



Hybrid encryption: pros and cons

Advantages:

- Key management in hybrid encryption is identical to key management in public-key cryptography (no shared secrets).
- Performance is close to symmetric-key.
- Security sometimes improves—hybrid encryption can be more secure than the cryptosystems you started with if combined carefully.

Disadvantages:

- Attack surface increases—if either the public-key or symmetric-key cryptosystem is totally broken, the hybrid encryption will be broken.

Hybrid encryption is used in:

- PGP, S/MIME ...
- and basically anything else that uses public-key encryption.

Equivalent security levels

Security in bits	Block cipher	Hash function	RSA/DH (bits)	ECC (bits)
80	SKIPJACK	(SHA-1)	1024	160
112	Triple-DES	SHA-224	2048	224
128	AES-128	SHA-256	3072	256
192	AES-192	SHA-384	7680	384
256	AES-256	SHA-512	15360	512

Basic hybrid encryption

- Let $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ be a public-key cryptosystem.
- Let (E, D) be a symmetric-key cryptosystem with ℓ-bit keys.
- Let $(k_{\mathrm{pubkey}}, k_{\mathrm{privkey}})$ be a public key/private key pair.
- Let m be a message.
- To perform hybrid encryption, choose $k \in \{0,1\}^\ell$ at random, and send

$$(c_1, c_2) = (\mathcal{E}(k_{\mathrm{pubkey}}, k), E(k, m))$$

- To decrypt (c_1, c_2), compute

$$m = D(\mathcal{D}(k_{\mathrm{privkey}}, c_1), c_2)$$

Security of hybrid encryption

Would like semantic security under adaptive chosen ciphertext attack (IND-CCA2).

Easy to show: if public-key cryptosystem and symmetric-key are IND-CCA2-secure, then basic hybrid encryption is IND-CCA2-secure.

Can we make IND-CCA2-secure hybrid encryption using weaker building blocks? Yes! See next few slides.

Improvements to basic hybrid encryption

Idea #1: Hash the key k before using it.

Encryption:

$$(c_1, c_2) = (\mathcal{E}(k_{\text{pubkey}}, k), E(H(k), m))$$

Decryption:

$$m = D(H(\mathcal{D}(k_{\text{privkey}}, c_1)), c_2)$$

Theorem (Kurosawa, Matsuo, ACISP 2004)

Hashed Elgamal hybrid encryption is semantically secure under adaptive chosen ciphertext attack (IND-CCA2), assuming:

- *the symmetric-key encryption scheme is semantically secure under adaptive chosen ciphertext attack (IND-CCA2),*
- *the hash function is a random oracle,*
- *the "Strong DH" problem is intractable.*

Diffie-Hellman Integrated Encryption Scheme (DHIES)

Idea #2: Add a MAC.

For example, Elgamal with a MAC:

Encryption: To encrypt m, choose r at random, and compute

$$(k_1, k_2) = H((g^\alpha)^r)$$
$$c = E(k_1, m)$$
$$t = \mathrm{MAC}(k_2, c)$$

The ciphertext is (g^r, c, t).

Decryption: Given a ciphertext (c_1, c_2, c_3), compute

$$(\hat{k}_1, \hat{k}_2) = H(c_1^\alpha)$$
$$\hat{m} = D(\hat{k}_1, c_2)$$
$$\hat{t} = \mathrm{MAC}(\hat{k}_2, c_2)$$

If $\hat{t} = c_3$, output $\hat{m}$, otherwise output NULL.

Diffie-Hellman Integrated Encryption Scheme (DHIES)

M. Abdalla, M. Bellare, and P. Rogaway, "The Oracle Diffie-Hellman Assumptions and an Analysis of DHIES," CT-RSA 2001, pp. 143–158.

- Also known as Diffie-Hellman Authenticated Encryption Scheme, DHAES, DHIES, or DLIES.
- DHIES is semantically secure under adaptive chosen ciphertext attack (IND-CCA2), assuming:
  - The symmetric-key encryption scheme is semantically secure under chosen plaintext attack (IND-CPA),
  - The MAC is secure (EUF-CMA),
  - The hash function is a random oracle, and
  - The Diffie-Hellman problem is intractable.

Note that hash+MAC achieves IND-CCA2 security, even though no underlying component encryption function is CCA2-secure.

Fujisaki-Okamoto cryptosystem

Idea #3: Instead of a MAC, a simple hash check is enough.

Key generation: Use $\mathcal{G}$ to generate public/private key pairs.

Encryption: To encrypt $m \in \{0,1\}^*$, compute

$$(c_1, c_2, c_3) = (\mathcal{E}(k_{\text{pubkey}}, k), E(H_1(k), m), H_2(m, k)),$$

for k chosen at random.

Decryption: To decrypt a ciphertext of the form (c_1, c_2, c_3):

$$\hat{k} = \mathcal{D}(k_{\text{privkey}}, c_1)$$

$$\hat{m} = D(H_1(\hat{k}), c_2)$$

$$\text{output} \begin{cases} \hat{m} & \text{if } c_3 = H_2(\hat{m}, \hat{k}) \\ \text{NULL} & \text{otherwise.} \end{cases}$$

E. Fujisaki and T. Okamoto, "Secure Integration of Asymmetric and Symmetric Encryption Schemes," CRYPTO 1999, pp. 537–554.

- The Fujisaki–Okamoto public-key cryptosystem is semantically secure under adaptive chosen ciphertext attack (IND-CCA2) if we assume:
  - The $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ public-key cryptosystem is one-way secure under chosen plaintext attack (OW-CPA),
  - The (E, D) symmetric-key encryption scheme is semantically secure under chosen plaintext attack (IND-CPA),
  - H_1 and H_2 are random oracles.
- The proof of security is easier if the (public-key) encryption function $\mathcal{E}$ is deterministic, but the result also holds for public-key cryptosystems with randomized $\mathcal{E}$.

Shoup's KEM/DEM approach

Standardized as ISO/IEC 18033-2 (2001)

- "Key encapsulation mechanism" (KEM):
  - Choose random $r \bmod pq$
  - Encrypt r with RSA ($c_1 = r^e \bmod pq$)
  - Set $k = H(r, c_1)$
- "Data encapsulation mechanism" (DEM)
  - Encrypt and authenticate m using AES-GCM with key k: $c_2 = \mathrm{AES\text{-}GCM}(k, m)$
  - Send c_1 and c_2
- To decrypt: Decrypt c_1, compute k, and decrypt c_2
- Provably secure, extremely efficient, and robust against design or implementation error