Public Key Encryption Based on Cyclic Groups

Palash Sarkar

Indian Statistical Institute

Structure of Presentation

- Conceptual overview and motivation.
- Basic Construction
  - Diffie-Hellman problem
  - ElGamal cryptosystem.
- PKE and Security Definitions.
- Hybrid encryption.
- Provable Constructions.
  - Cramer-Shoup and Kurosawa-Desmedt cryptosystems.
  - Recent work.

Conceptual Overview and Motivation

Science of Encryption

Evolution

- Classical cryptosystems.
  - encryption and decryption keys are same.
  - both are secret.
  - Problems: key distribution and management.
- Public key cryptosystems. A paradigm shift.
  - encryption and decryption keys are different.
  - encryption key is public; decryption key is secret.
  - Problems: Operational issues.

Public Key Encryption (PKE)

- Alice has two keys
  - pk_A : Available in a public directory.
  - sk_A : Kept secret by Alice.
- Bob encrypts a message using pk_A.
- Alice decrypts the ciphertext using sk_A.
- Problem: (Wo)man in the middle.
  - Eve impersonates Alice.
  - Puts a public key pk_E in Alice's name.
  - Eve decrypts any message encrypted using pk_E.

Digital Signature Protocol

- Consists of algorithms (Setup, Sign, Verify).
- Setup generates (pk_C, sk_C) for Charles.
- pk_C is made public (placed in a public directory).
- Charles signs message M using sk_C to obtain signature σ.
- Anybody can verify the validity of (M, σ) using pk_C.

Certifying Authority (CA)

- Consider Charles to be CA.
- Alice obtains certificate.
  - Alice generates (pk_A, sk_A); sends pk_A to CA.
  - CA signs (Alice, pk_A) using sk_C to obtain σ; Alice's certificate: (Alice, pk_A, σ).
- Bob sends message M to Alice.
  - Verifies (Alice, pk_A, σ) using pk_C.
  - Encrypts M using pk_A.

CA: Operational Issues

- How long will Alice's certificate be valid?
  - CA publishes certificate status information.
  - This information has to be fresh (to a day, for example).
  - Bob has to verify that Alice's certificate has not been revoked.
- Does Bob trust Alice's CA?
  - Alice and Bob may have different CAs.
  - This may lead to a chain (or tree) of CAs.
  - CAs have to certify each other.

Public Key Infrastructure

- Consists of certifying authorities and users.
- Certificate status information.
  - Certificate revocation list (CRL).
  - Online certificate status protocol (OCSP).
  - One-way hash chains.
- A major stumbling block for *widespread* adoption of PKE.

Basic Construction

Setting

Discrete Log Problem:

Instance: (g, h)

- $G = \langle g \rangle$ is a cyclic group.
- h is a random element of G.

Task: Compute $a = \log_g(h)$, i.e., a such that $h = g^a$.

Examples. A prime order subgroup of

- the multiplicative group of a finite field.
- the group of points of an elliptic curve over a finite field.
- the Jacobian of a hyperelliptic curve over a finite field.

Criteria

Suppose G is a subgroup of H.

Security:

- DLP should be computationally intractable.
- Possibly other problems should also be computationally intractable.
- The above determines $|G|$ and $|H|$.

Efficiency: Depends on

- $|G|$ and $|H|$.
- the time for one group operation in H;
- the time required to perform g^a.

Diffie-Hellman Problems

Computational Diffie-Hellman (CDH) problem:

Instance: (g, g^a, g^b)

- $G = \langle g \rangle$ is a cyclic group of order q;
- a, b are random elements of $\mathbb{Z}_q$.

Task: Compute g^{ab}.

Decision Diffie-Hellman (DDH) problem:

Instance: (g, g^a, g^b, h).

Task: Determine whether $h = g^{ab}$ or whether h is a random element of G.

Advantage

Let $\mathcal{A}$ be a probabilistic algorithm, which takes as input a tuple (g, g_1, g_2, g_3) and outputs a bit.

$$\begin{aligned}&\mathsf{Adv}_{\mathrm{DDH}}(\mathcal{A})\\ &\quad = |\mathsf{Pr}[\mathcal{A} \Rightarrow 1|(g, g_1, g_2, g_3) \text{ is real}]\\ &\qquad -\mathsf{Pr}[\mathcal{A} \Rightarrow 1|(g, g_1, g_2, g_3) \text{ is random}]|.\end{aligned}$$

(g, g_1, g_2, g_3) is real: $g_1 = g^a$, $g_2 = g^b$ and $g_3 = g^{ab}$, i.e., a proper DDH tuple.

(g, g_1, g_2, g_3) is random: g_1, g_2 and g_3 are random elements of G.

DDH is (t, ϵ)-hard: if for all $\mathcal{A}$ with run time at most t, $\mathsf{Adv}_{\mathrm{DDH}}(\mathcal{A}) \le \epsilon$.

DH Key Agreement

Set-Up: $G = \langle g \rangle$ is a cyclic group and $q = |G|$.

Alice	Bob
$r_A \xleftarrow{\$} \mathbb{Z}_q$	$r_B \xleftarrow{\$} \mathbb{Z}_q$
compute $h_A = g^{r_A}$	compute $h_B = g^{r_B}$
send h_A to Bob	send h_B to Alice
compute $K_{AB} = h_A^{r_B}$	compute $K_{AB} = h_B^{r_A}$

Public information: g, g^{r_A}, g^{r_B}.

Key: $g^{r_A r_B}$.

This protocol gives the CDH problem its name.

ElGamal Encryption

Set-Up: $G = \langle g \rangle$; $q = |G|$;

secret key $r_A \xleftarrow{\$} \mathbb{Z}_q$; public key $(g, h_A = g^{r_A})$.

Encryption. Input: message M.

$t \xleftarrow{\$} \mathbb{Z}_q$.

Compute $h = g^t$ and $K = h_A^t$.

"Mask" M using K to obtain C.

Send (h, C).

Decryption. Input: (h, C).

Compute $K = h^{r_A}$.

"Unmask" C using K to obtain M.

Comment: An implicit DH key agreement.

PKE and Security Definitions

PKE Definition

Consists of three probabilistic algorithms.

Set-Up. Input: a security parameter.

- Returns pk_A and sk_A of Alice.

Encrypt. message M; pk_A.

- Returns C to be the encryption of M under pk_A.

Encrypt. ciphertext C; pk_A; sk_A.

- Returns either
  - $\perp$ signifying that C is mal-formed; or
  - M.

Adversary Does What?

Intuitive goals of an adversary.

- Get the secret key of Alice.
- Try to decipher a ciphertext intended for Alice.
- Indistinguishability of ciphertexts.
  - Ask Alice to decrypt a few other (possibly mal-formed) ciphertexts.

Modelling Paranoid Security

- Adversarial goal: **Weak.**
  Two equal length messages M_0 and M_1 are produced by the adversary; a bit b is chosen and the adversary is given an encryption of M_b; adversary has to determine b.
  - Allowed to ask Alice for decryption of other ciphertexts.
- Adversarial resources: **maximum practicable.**
  Probabilistic algorithm.
  - Asymptotic setting: polynomial time (in the security parameter) computation.
    Concrete setting: relate success probability to running time.

Security Definition

Game between adversary and simulator.

Set-Up: simulator

- Generates (pk, sk).
- Provides pk to the adversary.
- Keeps sk secret.

Phase 1: adversarial queries.

- Decryption oracle: ask for the decryption of any ciphertext.

Security Definition (contd.)

Challenge:

- Adversary outputs two equal length messages M_0 and M_1.
- Simulator chooses a random bit b;
  encrypts M_b using pk to obtain C^*;
  gives C^* to the adversary.

Phase 2: adversarial queries.

- Restriction:
  cannot ask for the decryption of C^*.

Security Definition (contd.)

Guess:

- adversary outputs a bit b';
- adversary wins if $b = b'$.

Advantage:

$$\epsilon = |\Pr[b = b'] - 1/2|.$$

(ϵ, t)-adversary: running time t; advantage ϵ.

Security Definition (contd.)

- Strongest definition:
  security against adaptive chosen ciphertext attacks.
  CCA-secure (CCA2-secure).
- Weaker definition:
  Adversary not provided with the decryption oracle.
  security against chosen plaintext attacks.
  CPA-secure.

ElGamal is not CCA-Secure

Adversarial Steps.

- Set-Up: obtain $pk = g^r$ from the simulator.
- Phase 1: makes no queries.
- Challenge:
  provides two distinct group elements m_0 and m_1;
  obtains $(h = g^t, y = m_b \times g^{rt})$ in response.
- Phase 2: asks for decryption of (h, yz);
  receives $m_b z$ in response.
- Guess: computes $m_b = m_b z \times z^{-1}$;
  determines b with probability one.

Malleable. Convert a valid ciphertext into another valid ciphertext without knowing the secret key.

Hybrid Encryption

Some Efficiency Issues

Suppose G is a group of points obtained from a "suitable" elliptic curve.

- Encryption and decryption require several scalar multiplications.
- Each scalar multiplication requires several multiplications over the underlying finite field.
- Assuming encryption to be done block by block (which does not satisfy security definition), the time required will be large.

Symmetric Versus Asymmetric

- Most asymmetric encryption primitives require either a field exponentiation or a scalar multiplication.

  asymptotic complexity: $O(k^3)$, where k is a security parameter.
- Symmetric encryption primitives (block and stream ciphers) do not (usually) require field exponentiation or scalar multiplication.
- Consequence: symmetric encryption is much faster than asymmetric encryption.

Combine symmetric and asymmetric encryption to obtain the best of both worlds.

Hybrid Encryption – Basic Idea

Components.

Data Encapsulation Mechanism (DEM):
$\mathsf{Sym.Enc}_K()$ and $\mathsf{Sym.Dec}_K()$.

Key Encapsulation Mechanism (KEM):
$\mathsf{KEM.SetUp}()$, $\mathsf{KEM.Enc}()$ and $\mathsf{KEM.Dec}()$.

PKE Construction.

$\mathsf{PKE.SetUp}()$: $(pk, sk) = \mathsf{Asym.SetUp}()$.

$\mathsf{PKE.Enc}(pk, M)$:
$(A, K) = \mathsf{KEM.Enc}(pk)$; $B = \mathsf{Sym.Enc}_K(M)$;
return $C = (A, B)$.

$\mathsf{PKE.Dec}(pk, sk, C = (A, B))$:
$K = \mathsf{KEM.Dec}(pk, sk, A)$;
$M = \mathsf{Sym.Dec}_K(B)$.

Hybrid Encryption Issues

- Many details have been glossed over.
- Security.
  - CCA-secure KEM: definition similar to that of CCA-secure PKE.
  - CCA-secure DEM: definition based on the definition of security of symmetric encryption (not discussed here).
  - Generic security of hybrid PKE.
    CCA-secure KEM + CCA-secure DEM $\Rightarrow$ CCA-secure PKE.
- In special cases, the security conditions on either KEM or DEM can be relaxed.

Provable Constructions

What do we mean?

Construct a PKE such that one can *prove* that it satisfies the security definition.

Qualifiers.

- Proofs usually require an assumption.
  - Generic: (trapdoor) one-way functions exist.
  - Specific: the DDH problem is computationally intractable.
- Security statement: $\mathsf{Adv}_{pke} \leq f(\mathsf{Adv}_{\Pi})$ where Π is a computationally hard problem.
- Proofs are reductions. Transform a “successful” adversary for breaking PKE to a “good” algorithm for solving Π.

Constructions

- Cramer-Shoup (1998): based on hardness of DDH and *no other assumption.*
- Kurosawa-Desmedt (2004): A variant of Cramer-Shoup which performs more efficient hybrid encryption.
- Hofheinz-Kiltz (2007): based on hardness of a (possibly) weaker problem than DDH.
- Cash-Kiltz-Shoup (2008): based on twin Diffie-Hellman problem.
- Other constructions: require more assumptions.

Cramer-Shoup (1998)

Components.

- A cyclic group $G = \langle g \rangle$ of order q.
- A universal one-way hash family (UOWHF) $\{H\}_{s \in \mathcal{S}}$, where each $H_s : G^3 \rightarrow G$.
  The following game should be computationally hard.
  - Adversary outputs a.
  - Adversary is given $s \stackrel{\$}{\leftarrow} \mathcal{S}$.
  - Adversary has to output $a' \neq a$ such that $H(a) = H(a')$.

Cramer-Shoup (contd.)

SetUp.

- Choose $g_1, g_2 \xleftarrow{\$} G$.
- Choose $x_1, x_2, y_1, y_2, z \xleftarrow{\$} \mathbb{Z}_q$.
- Compute $c = g_1^{x_1} g_2^{x_2}$, $d = g_1^{y_1} g_2^{y_2}$, $h = g_1^z$.
- Choose $s \xleftarrow{\$} \mathcal{S}$ as key for H_s.
- Public key: (g_1, g_2, c, d, h, H).
- Secret key: (x_1, x_2, y_1, y_2, z).

Cramer-Shoup (contd.)

Encryption: message $m \in G$.

- Choose $r \xleftarrow{\$} G$.
- Compute $u_1 = g_1^r$, $u_2 = g_2^r$, $e = h^r m$.
- Compute $\alpha = H(u_1, u_2, e)$, $v = c^r d^{r\alpha}$.
- Ciphertext is (u_1, u_2, e, v).

Decryption: ciphertext (u_1, u_2, e, v).

- Compute $\alpha = H(u_1, u_2, e, v)$.
- Verify $u_1^{x_1+y_1\alpha} u_2^{x_2+y_2\alpha} \stackrel{?}{=} v$.
- If "not equal" output $\perp$ (reject).
- Else, output m/u_1^z.

Cramer-Shoup (contd.)

An alternative formulation of DDH.

Instance: (g_1, g_2, u_1, u_2).

Task: $\log_{g_1} u_1 \stackrel{?}{=} \log_{g_2} u_2$, i.e., whether there is an r such that $u_1 = g_1^r$ and $u_2 = g^r$.

Equivalence to DDH.

- $g_1 \longrightarrow g$, $g_2 \longrightarrow g^x$, $u_1 \longrightarrow g^y$, $u_2 \longrightarrow g^{xy}$.

Security of Cramer-Shoup PKE

Simulator SetUp.

- Input to simulator: (g_1, g_2, u_1, u_2).
- Simulator chooses $x_1, x_2, y_1, y_2, z_1, z_2 \in \mathbb{Z}_q$.
- Computes $c = g_1^{x_1} g_2^{x_2}$, $d = g_1^{y_1} g_2^{y_2}$, $h = g_1^{z_1} g_2^{z_2}$.
- Chooses $s \xleftarrow{\$} \mathcal{S}$.
- Outputs (g_1, g_2, c, d, h, H) as public key.
- Knows $(x_1, x_2, y_1, y_2, z_1, z_2)$.

Security of Cramer-Shoup PKE

Simulation of decryption oracle:

- As in the original protocol except for the following point.
- Computes $m = e/(u_1^{z_1} u_2^{z_2})$.

Simulation of challenge: input m_0, m_1

- $b \stackrel{\$}{\leftarrow} \{0, 1\}$.
- Computes $e = u_1^{z_1} u_2^{z_2} m_b$, $\alpha = H(u_1, u_2, e)$.
- Computes $v = u_1^{x_1 + y_1 \alpha} u_2^{x_2 + y_2 \alpha}$.
- Outputs (u_1, u_2, e, v).

Security of Cramer-Shoup PKE

- If the simulator's input is a random 4-tuple, then the bit b is statistically hidden from the adversary.
- If the simulator's input is a proper DH-tuple (as per the alternative formulation), then the simulation is perfect.
- A simple linear algebra argument is used to show that any invalid ciphertext is rejected by the simulator with overwhelming probability.

Summary

- An overview of PKE protocols.
- Framework in which they are used.
- Formal security model.
- A few constructions.
- A sketch of security proof of the Cramer-Shoup protocol.
- Pointers to more recent constructions.

Thank you for your kind attention!