Introduction to Cryptology

9.3 - Dlog, DHKE and ElGamal

Federico Pintore

Mathematical Institute, University of Oxford (UK)

UNIVERSITY OF OXFORD

Michaelmas term 2020

The Discrete Logarithm Problem (Dlog)

A group generation algorithm $\mathcal{G}$ is a PPT algorithm which:

- on input a security parameter n, outputs a description of a cyclic group $\mathbb{G}$, its order q and a generator $g \in \mathbb{G}$.
- $||q|| = \lfloor \log_2 q \rfloor + 1 = n$.
- Computing the group operation of $\mathbb{G}$ is efficient.

Given $h \in \mathbb{G}$, $\log_g h$ denotes the unique $x \in \{1, \ldots, q\}$ s.t. $h = g^x$.

The Discrete Logarithm Problem (Dlog)

A group generation algorithm $\mathcal{G}$ is a PPT algorithm which:

- on input a security parameter n, outputs a description of a cyclic group $\mathbb{G}$, its order q and a generator $g \in \mathbb{G}$.
- $||q|| = \lfloor \log_2 q \rfloor + 1 = n$.
- Computing the group operation of $\mathbb{G}$ is efficient.

Given $h \in \mathbb{G}$, $\log_g h$ denotes the unique $x \in \{1, \ldots, q\}$ s.t. $h = g^x$.

Discrete logarithm (Dlog) problem relative to $\mathcal{G}$: given $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(n)$ and a uniform $h \in \mathbb{G}$, compute x.

The Discrete Logarithm Problem (Dlog)

A group generation algorithm $\mathcal{G}$ is a PPT algorithm which:

- on input a security parameter n, outputs a description of a cyclic group $\mathbb{G}$, its order q and a generator $g \in \mathbb{G}$.
- $||q|| = \lfloor \log_2 q \rfloor + 1 = n$.
- Computing the group operation of $\mathbb{G}$ is efficient.

Given $h \in \mathbb{G}$, $\log_g h$ denotes the unique $x \in \{1, \ldots, q\}$ s.t. $h = g^x$.

Discrete logarithm (Dlog) problem relative to $\mathcal{G}$: given $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(n)$ and a uniform $h \in \mathbb{G}$, compute x.

The Dlog problem is hard relative to $\mathcal{G}$ if, for every PPT adversary $\mathcal{A}$, their success probability is negligible in n.

Diffie-Hellman Problem and its variants

Computational Diffie-Hellman (CDH) problem relative to $\mathcal{G}$: given $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(n)$ and two uniform $h, k \in \mathbb{G}$, where $h_1 = g^x$ and $h_2 = g^y$, compute g^{xy}.

- If the Dlog problem is easy, also the CDH problem is.
- The reverse implication is not clear.

Diffie-Hellman Problem and its variants

Computational Diffie-Hellman (CDH) problem relative to $\mathcal{G}$: given $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(n)$ and two uniform $h, k \in \mathbb{G}$, where $h_1 = g^x$ and $h_2 = g^y$, compute g^{xy}.

- If the Dlog problem is easy, also the CDH problem is.
- The reverse implication is not clear.

Decisional Diffie-Hellman (DDH) problem relative to $\mathcal{G}$: given $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(n)$, two uniform $h_1, h_2 \in \mathbb{G}$, where $h_1 = g^x$ and $h_2 = g^y$, and a third element z, decide if $z = g^{xy}$ or it is a uniform group element.

- If the CDH problem is easy, then also the DDH problem is.
- The reverse implication does not appear to be true.

Gap Diffie-Hellman $\mathcal{G}$

Definition

A group generation algorithm $\mathcal{G}$ is gap-DH if the DDH problem relative to $\mathcal{G}$ is easy but the CDH problem is still hard.

There exist concrete group generation algorithms that are gap-DH.

Diffie-Hellman Key-Exchange

Public parameters: $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(n)$.

- Alice chooses a uniform $a \in \{1, \ldots, q\}$, and sends $h_A = g^a$ to Bob.
- Bob chooses a uniform $b \in \{1, \ldots, q\}$, and sends $h_B = g^b$ to Alice.
- Alice computes $(g^b)^a = g^{ab}$.
- Bob computes $(g^a)^b = g^{ab}$.

Diffie-Hellman security

The security follows ***almost*** directly from the hardness of the DDH problem relative to $\mathcal{G}$.

The hardness of the Dlog problem is necessary for the security of the Diffie-Hellman key-exchange, but it may not be sufficient.

ElGamal Encryption Scheme

The ElGamal public-key encryption scheme $(\mathrm{KeyGen}, \mathrm{Enc}, \mathrm{Dec})$ relative to a group generation algorithm $\mathcal{G}$ is defined as follows:

- $(\mathrm{PK}, \mathrm{SK}) \leftarrow \mathrm{KeyGen}(1^n)$: on input a security parameter n, it runs $\mathcal{G}$ on n, obtaining a description of a cyclic group $\mathbb{G}$ - having order q, with $||q|| = n$ - together with a generator g.

  It picks a uniform $x \in \{1, \ldots, q\}$ and computes $h \leftarrow g^x$. The public key is $\mathrm{PK} = (\mathbb{G}, g, q, h)$ and the secret key is $\mathrm{SK} = x$.

- $c \leftarrow \mathrm{Enc}(\mathrm{PK}, m \in \mathbb{G})$: given a public key PK and a message m, it chooses a uniform $y \in \mathbb{Z}_q$, and outputs

$$c = (c_1, c_2) := (g^y, h^y \cdot m).$$

- $m \leftarrow \mathrm{Dec}(\mathrm{SK}, c)$: on input a secret key $\mathrm{SK} = x$ and a ciphertext $c = (c_1, c_2)$, it outputs $m = c_2 / c_1^x$.

Correctness: $c_2/c_1^x = h^y \cdot m/(g^y)^x = (g^x)^y \cdot m/(g^y)^x = m$.

ElGamal Encryption Scheme

Lemma

Let $\mathbb{G}$ be a finite group. If an arbitrary element $m \in \mathbb{G}$ is multiplied by an uniform group element $k \in \mathbb{G}$, the result $k \cdot m$ is a uniform group element as well.

ElGamal Encryption Scheme

Lemma

Let $\mathbb{G}$ be a finite group. If an arbitrary element $m \in \mathbb{G}$ is multiplied by an uniform group element $k \in \mathbb{G}$, the result $k \cdot m$ is a uniform group element as well.

Proof.

Given $g \in \mathbb{G}$, we have

$$\Pr(k \cdot m = g) = \Pr(k = g \cdot m^{-1}).$$

Because k is uniform, we obtain

$$\Pr(k = g \cdot m^{-1}) = 1/|\mathbb{G}|.$$

□

Security of the ElGamal Encryption Scheme

Theorem

If the DDH problem is hard relative to $\mathcal{G}$, then the ElGamal encryption scheme relative to $\mathcal{G}$ is CPA-secure.

Proof.

Let $\mathcal{A}$ a PPT adversary against the ElGamal encryption scheme, which we denote by S.

$\mathcal{A}$ is used as a subroutine to construct a PPT distinguisher D against the DDH problem relative to $\mathcal{G}$.

D receives an instance of the DDH problem, i.e.

$$(\mathbb{G}, q, g, h_1 = g^x, h_2 = g^y, z),$$

and it has to determine if $z = g^{xy}$ or $z = g^w$ for a uniform $w \in \{1, \ldots, q\}$.

Security of ElGamal Encryption Scheme

D works as follows:

- It sets $\mathrm{PK} = (\mathbb{G}, q, g, h_1)$ and sends it to $\mathcal{A}$.
- Upon reception of (m_0, m_1) from $\mathcal{A}$, D picks $b \in \{0, 1\}$, sets $c_1 = h_2$ and $c_2 = z \cdot m_b$, and sends $c = (c_1, c_2)$ to $\mathcal{A}$.
- It outputs 1 if the bit b' received from $\mathcal{A}$ is equal to b, 0 otherwise.

Security of ElGamal Encryption Scheme

D works as follows:

- It sets $\text{PK} = (\mathbb{G}, q, g, h_1)$ and sends it to $\mathcal{A}$.
- Upon reception of (m_0, m_1) from $\mathcal{A}$, D picks $b \in \{0, 1\}$, sets $c_1 = h_2$ and $c_2 = z \cdot m_b$, and sends $c = (c_1, c_2)$ to $\mathcal{A}$.
- It outputs 1 if the bit b' received from $\mathcal{A}$ is equal to b, 0 otherwise.

Let S' be a modified version of ElGamal, where Enc chooses uniform $y, w \in \{1, \ldots, q\}$, and outputs $c = (g^y, g^w \cdot m)$.

S' does not satisfy correctness, but the game $\text{PubK}^{\text{eav}}_{\mathcal{A},S'}(n) = 1$ is still well-defined.

Security of ElGamal Encryption Scheme

Since c_2 is a uniformly distributed group element, it holds

$$\Pr(\mathrm{PubK}^{\mathrm{eav}}_{\mathcal{A},S'}(n) = 1) = 1/2$$

Security of ElGamal Encryption Scheme

Since c_2 is a uniformly distributed group element, it holds

$$\Pr(\mathrm{PubK}^{\mathrm{eav}}_{\mathcal{A},S'}(n) = 1) = 1/2$$

Case 1 - random tuple: the view of $\mathcal{A}$ when run as a subroutine by D is distributed identically to their view in $\mathrm{PubK}^{\mathrm{eav}}_{\mathcal{A},S'}$. Hence:

$$\Pr(\mathrm{D}(\mathbb{G}, q, g, g^x, g^y, g^w) = 1) = \Pr(\mathrm{PubK}^{\mathrm{eav}}_{\mathcal{A},S'}(n) = 1) = 1/2 \quad (1)$$

Security of ElGamal Encryption Scheme

Case 2 - DH tuple: the view of $\mathcal{A}$ when run as a subroutine by D is distributed identically to their view in $\mathrm{PubK}^{\mathrm{eav}}_{\mathcal{A},S}$. Therefore

$$\Pr(D(\mathbb{G}, q, g, g^x, g^y, g^{xy}) = 1) = \Pr(\mathrm{PubK}^{\mathrm{eav}}_{\mathcal{A},S}(n) = 1) \quad (2)$$

Security of ElGamal Encryption Scheme

Case 2 - DH tuple: the view of $\mathcal{A}$ when run as a subroutine by D is distributed identically to their view in $\mathrm{PubK}^{\mathrm{eav}}_{\mathcal{A},\mathcal{S}}$. Therefore

$$\Pr(\mathrm{D}(\mathbb{G}, q, g, g^x, g^y, g^{xy}) = 1) = \Pr(\mathrm{PubK}^{\mathrm{eav}}_{\mathcal{A},\mathcal{S}}(n) = 1) \tag{2}$$

If the DDH problem is hard relative to $\mathcal{G}$, then

$$\begin{aligned} |\Pr(\mathrm{D}(\mathbb{G}, q, g, g^x, g^y, g^w) = 1) - \\ \Pr(\mathrm{D}(\mathbb{G}, q, g, g^x, g^y, g^{xy}) = 1)| \leq \mathrm{negl}(n) \end{aligned} \tag{3}$$

Security of ElGamal Encryption Scheme

Case 2 - DH tuple: the view of $\mathcal{A}$ when run as a subroutine by D is distributed identically to their view in $\mathrm{PubK}^{\mathrm{eav}}_{\mathcal{A},S}$. Therefore

$$\Pr(\mathrm{D}(\mathbb{G}, q, g, g^x, g^y, g^{xy}) = 1) = \Pr(\mathrm{PubK}^{\mathrm{eav}}_{\mathcal{A},S}(n) = 1) \qquad (2)$$

If the DDH problem is hard relative to $\mathcal{G}$, then

$$\begin{aligned} |\Pr(\mathrm{D}(\mathbb{G}, q, g, g^x, g^y, g^w) = 1) - \\ \Pr(\mathrm{D}(\mathbb{G}, q, g, g^x, g^y, g^{xy}) = 1)| \le \mathrm{negl}(n) \end{aligned} \qquad (3)$$

From equations (1), (2) and (3) we deduce

$$\Pr(\mathrm{PubK}^{\mathrm{eav}}_{\mathcal{A},S}(n)) \le 1/2 + \mathrm{negl}(n)\,.$$

□

ElGamal Encryption Scheme - CCA-secure?

The ElGamal encryption scheme is malleable, hence it is not CCA-secure (CCA-secure schemes are non-malleable).

Malleability: given a ciphertext c, which is the encryption of an unknown message m, it is possible to generate an encryption c' of a message m' which has some known relation with m.

ElGamal Encryption Scheme - CCA-secure?

The ElGamal encryption scheme is malleable, hence it is not CCA-secure (CCA-secure schemes are non-malleable).

Malleability: given a ciphertext c, which is the encryption of an unknown message m, it is possible to generate an encryption c' of a message m' which has some known relation with m.

- Consider PK$= (\mathbb{G}, q, g, h)$ and the encryption (c_1, c_2) of a message m.
- In the modification $(c_1, c_2' = \alpha \cdot c_2)$, where $\alpha \in \mathbb{G}$, we have $c_1 = g^y$ and $c_2' = h^y \cdot \alpha \cdot m$.
- Hence (c_1, c_2') is a valid encryption of $\alpha \cdot m$.

A CPA-secure KEM based on DDH

Consider the following key-encapsulation mechanism (KeyGen, Encaps, Decaps) relative to a group generation algorithm $\mathcal{G}$:

- $(\mathrm{PK}, \mathrm{SK}) \leftarrow \mathrm{KeyGen}(n)$: it runs $\mathcal{G}$ on a security parameter n to generate $(\mathbb{G}, q, g)$. It then samples a uniform $x \in \{1, \ldots, q\}$, computes $h = g^x$ and specifies a hash function $H : \mathbb{G} \to \{0,1\}^{\ell(n)}$.

  The public key is $\mathrm{PK} = (\mathbb{G}, q, g, h, H)$, the private key is x.
- $(c, k) \leftarrow \mathrm{Encaps}(\mathrm{PK}, n)$: on input a public key PK and a security parameter n, it chooses a uniform $y \in \{1, \ldots, 1\}$ and outputs the ciphertext $c := g^y$ and the key $H(h^y)$.
- $k \leftarrow \mathrm{Decaps}(\mathrm{SK}, c)$: on input a secret key $\mathrm{SK} = x$ and a ciphertext c, it outputs $H(c^x)$.

A CPA-secure KEM based on DDH

Consider the following key-encapsulation mechanism (KeyGen, Encaps, Decaps) relative to a group generation algorithm $\mathcal{G}$:

- $(\mathrm{PK}, \mathrm{SK}) \leftarrow \mathrm{KeyGen}(n)$: it runs $\mathcal{G}$ on a security parameter n to generate $(\mathbb{G}, q, g)$. It then samples a uniform $x \in \{1, \ldots, q\}$, computes $h = g^x$ and specifies a hash function $H : \mathbb{G} \to \{0,1\}^{\ell(n)}$.

  The public key is $\mathrm{PK} = (\mathbb{G}, q, g, h, H)$, the private key is x.
- $(c, k) \leftarrow \mathrm{Encaps}(\mathrm{PK}, n)$: on input a public key PK and a security parameter n, it chooses a uniform $y \in \{1, \ldots, 1\}$ and outputs the ciphertext $c := g^y$ and the key $H(h^y)$.
- $k \leftarrow \mathrm{Decaps}(\mathrm{SK}, c)$: on input a secret key $\mathrm{SK} = x$ and a ciphertext c, it outputs $H(c^x)$.

If H is modelled as a random oracle and the CDH problem relative to $\mathcal{G}$ is hard, then the above KEM is CPA-secure.

Further Reading I

- Mihir Bellare, Alexandra Boldyreva, and Silvio Micali.
  Public-key encryption in a multi-user setting: Security proofs and improvements.
  In Bart Preneel, editor, Advances in Cryptology — EUROCRYPT 2000, volume 1807 of Lecture Notes in Computer Science, pages 259–274. Springer Berlin Heidelberg, 2000.

- Dan Boneh.
  Simplified OAEP for the RSA and Rabin Functions.
  In Joe Kilian, editor, Advances in Cryptology — CRYPTO 2001, volume 2139 of Lecture Notes in Computer Science, pages 275–291. Springer Berlin Heidelberg, 2001.

Further Reading II

Ronald Cramer and Victor Shoup.
Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack.
SIAM Journal on Computing, 33(1):167–226, 2003.

Whitfield Diffie and Martin E Hellman.
New directions in cryptography.
Information Theory, IEEE Transactions on, 22(6):644–654, 1976.

Itai Dinur, Orr Dunkelman, Nathan Keller, and Adi Shamir.
New attacks on feistel structures with improved memory complexities.
In Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I, pages 433–454, 2015.

Further Reading III

Naofumi Homma, Atsushi Miyamoto, Takafumi Aoki, Akashi Satoh, and Adi Shamir.
Collision-based power analysis of modular exponentiation using chosen-message pairs.
In Cryptographic Hardware and Embedded Systems - CHES 2008, 10th International Workshop, Washington, D.C., USA, August 10-13, 2008. Proceedings, pages 15–29, 2008.