Cryptography

Discrete-Log Based Public Key Encryption

University of Birmingham

Autumn Term 2017

Lecturer: David Galindo

UNIVERSITY OF BIRMINGHAM | Security and Privacy

Let $G = \mathbb{Z}_{11}^{\star} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. Then $\text{ord}(G) = 10$

i	0	1	2	3	4	5	6	7	8	9	10
$2^i \mod 11$	1	2	4	8	5	10	9	7	3	6	1
$5^i \mod 11$	1	5	3	4	9	1	5	3	4	9	1

Let $G = \mathbb{Z}_{11}^{\star} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. Then $\text{ord}(G) = 10$

i	0	1	2	3	4	5	6	7	8	9	10
$2^i \mod 11$	1	2	4	8	5	10	9	7	3	6	1
$5^i \mod 11$	1	5	3	4	9	1	5	3	4	9	1

$$\begin{aligned} \langle 2 \rangle &= \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \\ \langle 5 \rangle &= \{1, 3, 4, 5, 9\} \end{aligned}$$

Let $G = \mathbb{Z}_{11}^{\star} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. Then $\text{ord}(G) = 10$

i	0	1	2	3	4	5	6	7	8	9	10
$2^i \mod 11$	1	2	4	8	5	10	9	7	3	6	1
$5^i \mod 11$	1	5	3	4	9	1	5	3	4	9	1

$$\begin{aligned} \langle 2 \rangle &= \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \\ \langle 5 \rangle &= \{1, 3, 4, 5, 9\} \end{aligned}$$

- 2 is called a **generator** of $\mathbb{Z}_{11}^{\star}$ because $\langle 2 \rangle = \mathbb{Z}_{11}^{\star}$
- 5 is not a generator of $\mathbb{Z}_{11}^{\star}$ because $\langle 5 \rangle \neq \mathbb{Z}_{11}^{\star}$
- $\mathbb{Z}_{11}^{\star}$ **is cyclic** because it has a generator

The multiplicative group $\mathbb{Z}_p^\star$

- Let p be a prime integer
  - The set $\mathbb{Z}_p^\star = \{1, \ldots, p-1\}$ is the set of integers modulo p which are invertible modulo p
- Generators of $\mathbb{Z}_p^\star$:
  - $\mathbb{Z}_p^\star$ is a **cyclic group** for multiplication modulo p, namely, there exists $g \in \mathbb{Z}_p^\star$ (called **generator**), such that

    $$\mathbb{Z}_p^\star = \{g^{p-1} = 1, g, g^2, \ldots, g^{p-2}\}$$

    We denote this by $\mathbb{Z}_p^\star = \langle g \rangle$
  - Any $h \in \mathbb{Z}_p^\star$ can be uniquely written as $h = g^x \mod p$ with $0 \leq x < p-1$; equivalently $x \in \mathbb{Z}_{p-1}$
  - The integer x is called the **discrete logarithm** of h to the base g, and denoted $\log_g h$ or $\mathrm{DLog}_g(h)$
  - The order of a group G is its number of elements, denoted $|G|$ or $\mathrm{ord}(G)$

Example

Let $G = \mathbb{Z}_{11}^{\star} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. We have seen that 2 is a generator, so $\mathrm{DLog}_2(a) = i$ means that i is the exponent $i \in \mathbb{Z}_{10}$ such that $2^i \mod 11 = a$

i	0	1	2	3	4	5	6	7	8	9
$2^i \mod 11$	1	2	4	8	5	10	9	7	3	6

a	1	2	3	4	5	6	7	8	9	10
$\mathrm{DLog}_2(a)$	0	1	8	2	4	9	7	3	6	5

i	0	1	2	3	4	5	6	7	8	9
$7^i \mod 11$	1	7	5	2	3	10	4	6	9	8

a	1	2	3	4	5	6	7	8	9	10
$\mathrm{DLog}_7(a)$										

Example

Let $G = \mathbb{Z}_{11}^{\star} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. We have seen that 2 is a generator, so $\mathrm{DLog}_2(a) = i$ means that i is the exponent $i \in \mathbb{Z}_{10}$ such that $2^i \mod 11 = a$

i	0	1	2	3	4	5	6	7	8	9
$2^i \mod 11$	1	2	4	8	5	10	9	7	3	6

a	1	2	3	4	5	6	7	8	9	10
$\mathrm{DLog}_2(a)$	0	1	8	2	4	9	7	3	6	5

i	0	1	2	3	4	5	6	7	8	9
$7^i \mod 11$	1	7	5	2	3	10	4	6	9	8

a	1	2	3	4	5	6	7	8	9	10
$\mathrm{DLog}_7(a)$	0	3	4	6	2	7	1	9	8	5

Finding a generator of $\mathbb{Z}_p^\star$

- Finding a generator g of $\mathbb{Z}_p^\star$ for prime p
  - The factorization of $p-1$ is needed. Otherwise, no efficient algorithm is known
  - Factoring is hard, but it is possible to generate p such that the factorization of $p-1$ is known
- Generator of $\mathbb{Z}_p^\star$
  - $g \in \mathbb{Z}_p^\star$ is a generator of $\mathbb{Z}_p^\star$ if and only if $g^{(p-1)/q} \neq 1 \mod p$ for each prime factor q of $p-1$
  - There are $\phi(p-1)$ generators of $\mathbb{Z}_p^\star$

Discrete logarithm problems

- **Discrete logarithm** (DL) problem:
  - Given $\mathbb{Z}_p^\star = \langle g \rangle$ and $g, h \in \mathbb{Z}_p^\star$ with $h = g^x \mod p$ for x random in $\mathbb{Z}_{p-1}$, compute $x = \mathrm{DLog}_g(h)$
- **Computational Diffie-Hellman** (CDH) problem:
  - Given $\mathbb{Z}_p^\star = \langle g \rangle$ and $g, h_1 = g^{x_1}, h_2 = g^{x_2} \in \mathbb{Z}_p^\star$ compute $g^{x_1 x_2} \mod p$
- **Decisional Diffie-Hellman** in a group G_q with $q|p-1$
  - Given $\mathbb{Z}_p^\star = \langle g \rangle$, for x, y, z random in $\mathbb{Z}_q$, where $G_q = \langle \hat{g} \rangle$, distinguish the tuples $(\hat{g}, \hat{g}^x, \hat{g}^y, \hat{g}^z)$ and $(\hat{g}, \hat{g}^x, \hat{g}^y, \hat{g}^{xy})$, where all exponentiations are $\mod p$

Discussion on CDH and DDH

Let G_q be a subgroup of $\mathbb{Z}_p^*$

If the Decisional Diffie-Hellman problem is hard to solve in G_q, this means that we cannot check whether a given element T in G_q is a solution of $\mathrm{CDH}_{G_q,g}(\cdot,\cdot)$

$$\begin{aligned} \mathrm{CDH}_{G_q,g} \quad : \quad & G_q \times G_q \to G_q \\ & (g^a, g^b) \mapsto g^{ab} \end{aligned}$$

i.e, whether $\mathrm{CDH}_{G_q,g}(g^a, g^b) = T$ without knowing a nor b

Hardness of Discrete Logarithm problems

- Computing discrete logarithms in $\mathbb{Z}_p^\star$
  - Hard problem: no efficient algorithm is known for large p
  - Brute force: enumerate all possible x. Complexity $\mathcal{O}(p)$
  - Baby step/giant step method: complexity $\mathcal{O}(\sqrt{p})$
- Computing CDH in $\mathbb{Z}_p^\star$
  - Hard problem: no efficient algorithm is known for large p
  - Best method: computing discrete logarithms
- Solving DDH in a subgroup $G_q \subset \mathbb{Z}_p^\star$ with $q|p-1$
  - Hard problem: no efficient algorithm is known for large p, q
  - Best method: computing CDH

Sophie Germain

(1776-1831)

Mathematician

A challenge was issued in Napoleonic France to explain why sand on small glass plates settled into patterns when the glass was vibrated. The only entrant was Sophie Germain. It took her six years, but she eventually won with a pioneering paper on elasticity. Despite her work, she was never accepted by the male establishment of the time.

Special case: Sophie Germain primes q

If both q and $2q + 1$ are prime, then q is a **Sophie Germain prime**

- We want to work in a prime-order subgroup of $\mathbb{Z}_p^\star$
  - Generate p, q such that $p - 1 = 2 \cdot q$ and p, q are prime
  - Find a generator g of $\mathbb{Z}_p^\star$. Let $\hat{g} = g^2 \mod p$ and $G_q = \langle \hat{g} \rangle$
  - G_q is the *Quadratic Residues* subgroup of $\mathbb{Z}_p^\star$
- Generate p such that $p - 1 = 2 \cdot q$ for some prime q
  - Generate a random prime p
  - Test if $q = (p - 1)/2$ is prime. If not, generate another p

Example

i	0	1	2	3	4	5	6	7	8	9	10
$7^i \mod 11$	1	7	5	2	3	10	4	6	9	8	1

i	0	1	2	3	4	5	6	7	8	9	10
$5^i \mod 11$	1	5	3	4	9	1	5	3	4	9	1

- Example
  - Let $p = 11, q = 5$, and $\mathbb{Z}_{11}^{\star} = \langle 7 \rangle$
  - $\hat{g} = 7^2 = 5 \mod 11$ and then $G_5 = \langle 5 \rangle = \{1, 3, 4, 5, 9\}$
  - G_5 is the group of Quadratic Residues modulo 11
  - $(7, 3, 5, 9)$ is a DH-tuple (g, g^x, g^y, g^{xy}):
    - $(g := 7, 7^4 = 3, 7^2 = 5, 7^{4 \cdot 2} = 9)$
  - $(5, 3, 4, 9)$ is not a DH-tuple $(\hat{g}, \hat{g}^x, \hat{g}^y, \hat{g}^{xy})$:
    - $(\hat{g} := 5, 5^2 = 3, 5^3 = 4, 5^{2 \cdot 3} = 5)$

ElGamal encryption (Taher ElGamal, 1985)

- Key generation
  - Let $p = 2 \cdot q + 1$ be a prime wrt. security parameter λ
  - Let g be such that $g^q = 1 \mod p$
  - Let G_q be the subgroup of $\mathbb{Z}_p^\star$ generated by g
  - Let $x \xleftarrow{R} \mathbb{Z}_q$. Let $h = g^x \mod p$
  - Public-key : (G, g, h). Private-key : x
- Encryption of $m \in G_q$:
  - Let $r \xleftarrow{R} \mathbb{Z}_q$
  - Output $c = (g^r, h^r \cdot m)$
- Decryption of $c = (c_1, c_2)$
  - Output $m = c_2 \cdot (c_1^x)^{-1} \mod p$

Furthermore, $\mathrm{Dec}(SK, \mathrm{Enc}(PK, m)) = m$ for $(PK, SK) \leftarrow \mathrm{KG}(\lambda)$

Key sizes (NIST 2016 recommendations)

Date	Minimum of Strength	Symmetric Algorithms	Factoring Modulus	Discrete Logarithm		Elliptic Curve	Hash (A)	Hash (B)
				Key	Group			
(Legacy)	80	2TDEA*	1024	160	1024	160	SHA-1**	
2016 - 2030	112	3TDEA	2048	224	2048	224	SHA-224 SHA-512/224 SHA3-224	
2016 - 2030 & beyond	128	AES-128	3072	256	3072	256	SHA-256 SHA-512/256 SHA3-256	SHA-1
2016 - 2030 & beyond	192	AES-192	7680	384	7680	384	SHA-384 SHA3-384	SHA-224 SHA-512/224
2016 - 2030 & beyond	256	AES-256	15360	512	15360	512	SHA-512 SHA3-512	SHA-256 SHA-512/256 SHA-384 SHA-512 SHA3-512

https://www.keylength.com/

Security of ElGamal: one-wayness

- **[Breaking one-wayness]** To recover m from $(g^r, h^r \cdot m)$
  - One must find h^r from $(g, g^r, h = g^x)$
  - Equivalently, one must find $g^{x \cdot r}$ from (g, g^r, g^x)
- Computational Diffie-Hellmann problem (CDH):
  - Given (g, g^a, g^b), find g^{ab}
  - No efficient algorithm is known
  - Best algorithm is finding the discrete-log
- **ElGamal is one-way** if and only if **solving CDH is infeasible** (i.e. the probability of succeeding in solving CDH is *negligible* in the security parameter λ)

Security of ElGamal: one-wayness

- **[Breaking one-wayness]** To recover m from $(g^r, h^r \cdot m)$
  - One must find h^r from $(g, g^r, h = g^x)$
  - Equivalently, one must find $g^{x \cdot r}$ from (g, g^r, g^x)
- Computational Diffie-Hellmann problem (CDH):
  - Given (g, g^a, g^b), find g^{ab}
  - No efficient algorithm is known
  - Best algorithm is finding the discrete-log
- **ElGamal is one-way** if and only if **solving CDH is infeasible** (i.e. the probability of succeeding in solving CDH is *negligible* in the security parameter λ)
- However, attacker may already have some information about the plaintext!

Security of ElGamal: semantic security

- Indistinguishability of encryption (IND-CPA)
  - The attacker receives pk
  - The attacker outputs two messages m_0, m_1
  - The attacker receives encryption of m_β for random bit β
  - The attacker outputs a "guess" β' of β
- Adversary's advantage
  - $\mathsf{Adv} = |\Pr[\beta' = \beta] - \frac{1}{2}|$
- **ElGamal is IND-CPA** (semantically secure) if and only if **solving Decisional DH is infeasible** (i.e. the probability of succeeding in solving DDH is negligible in the security parameter λ)

ElGamal is multiplicative

Given:

- $\text{Enc}(PK, m) = (c_1, c_2) = (g^r, h^r \cdot m)$
- $\text{Enc}(PK, m') = (c_1', c_2') = (g^{r'}, h^{r'} \cdot m')$
- Then one can obtain an encryption of $m \cdot m'$ by "multiplying" the previous ciphertexts:
  $\text{Enc}(PK, m \cdot m') = (c_1 \cdot c_1', c_2 \cdot c_2') = (g^{r+r'}, h^{r+r'} \cdot m \cdot m')$
- This is a very useful feature (e.g. e-voting, e-auctions, proxy re-encryption)
- ... but it can also lead to vulnerabilities

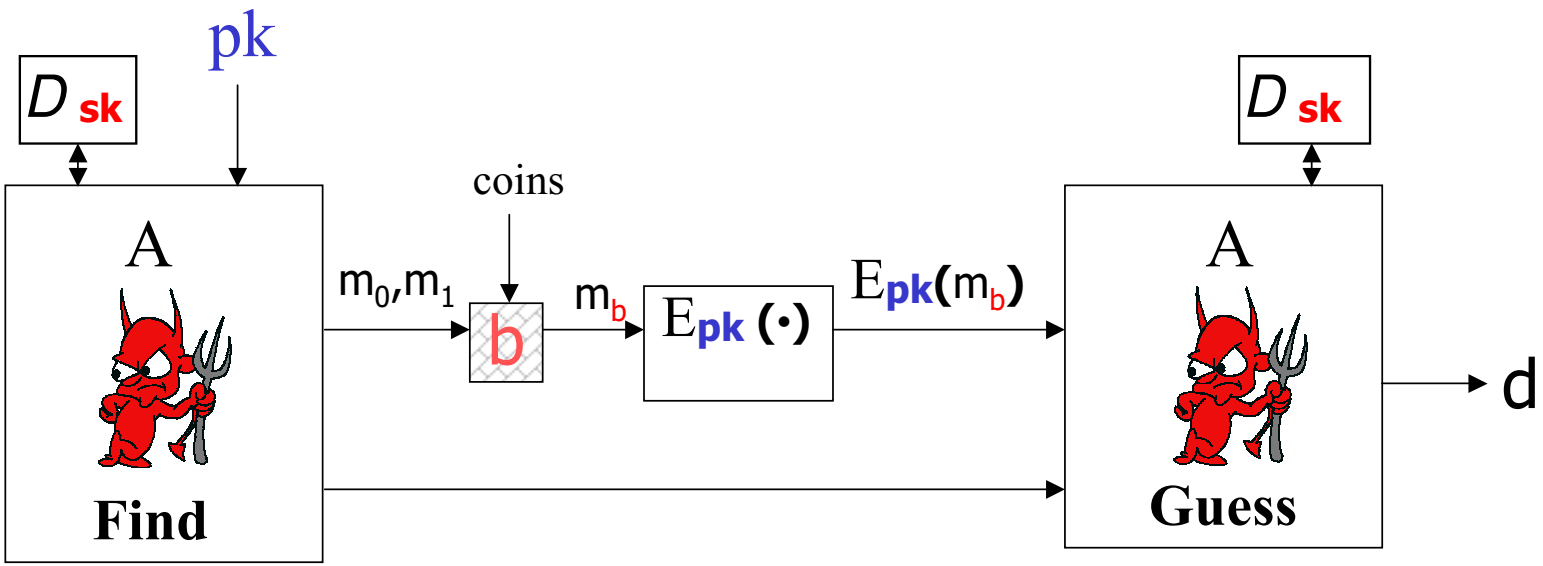
Definition (Chosen-ciphertext security)

Indistinguishability under chosen-ciphertext attack (IND-CCA) game is played between the challenger and an attacker

- The challenger runs $(PK, SK) \leftarrow \mathsf{KG}(\lambda)$ and passes the public key PK to the attacker
- The attacker is given access to a decryption oracle that answers $\mathsf{Dec}(SK, C)$ on input a ciphertext C
- The attacker submits two messages m_0 and m_1 of equal length to the challenger
- The challenger selects a bit $\beta \in \{0, 1\}$ at random
- The challenger returns $C_\beta = \mathsf{Enc}(PK, m_\beta)$ to the attacker
- The attacker gets help from the decryption oracle $\mathsf{Dec}(SK, C)$ on inputs $C \neq C_\beta$, and outputs a bit β'

The attacker wins this game if $\beta' = \beta$

IND-CCA game



Chosen-ciphertext attack

- El-Gamal is not chosen-ciphertext secure
  - Given $c = (g^r, h^r \cdot m)$ where $pk = (g, h)$
  - Ask for the decryption of $c' = (g^{r+1}, h^{r+1} \cdot m)$ and recover m

Chosen-ciphertext attack

- El-Gamal is not chosen-ciphertext secure
  - Given $c = (g^r, h^r \cdot m)$ where $pk = (g, h)$
  - Ask for the decryption of $c' = (g^{r+1}, h^{r+1} \cdot m)$ and recover m
- Sketch of proof: **ElGamal is not IND-CCA secure**
  - Attacker receives $pk = (p, q, G_q, g, h)$
  - Attacker chooses $m_0, m_1 \in G_q$ s.t. $m_0 \neq m_1$
  - Challenger outputs $C_\beta = (c_1, c_2) = (g^r, h^r \cdot m_\beta)$ for $\beta \in \{0, 1\}$ random
  - Attacker submits $C'_\beta := (g \cdot c_1, h \cdot c_2)$ to the decryption oracle
  - Obviously $C'_\beta \neq C_\beta$
  - Oracle returns $\mathrm{Dec}(sk, C'_\beta) = m_b$ since C'_β is an encryption of m_β
  - Attacker returns $\beta' = 0$ if $m_\beta = m_0$; returns $\beta' = 1$ otherwise
  - $\Pr[\beta = \beta'] - 1/2 = 1 - 1/2 = 1/2$

Twin ElGamal encryption (Cash, Kiltz, Shoup 2008)

We need:

- G_q a q prime-order subgroup of $\mathbb{Z}_p^\star$
- symmetric cipher (E, D)
- a message authentication code MAC
- a hash function $H : \{0,1\}^\star \to \mathcal{K}_{enc} \times \mathcal{K}_{mac}$

Twin ElGamal encryption (Cash, Kiltz, Shoup 2008)

- Key generation (on input p, q, G_q)
  - Let g be a generator of G_q
  - Let $x_1, x_2 \stackrel{R}{\leftarrow} \mathbb{Z}_q$, and $Y_1 = g^{x_1} \mod p$, $Y_2 = g^{x_2} \mod p$
  - Public-key (p, q, G_q, g, Y_1, Y_2) Private-key (x_1, x_2)
- Encryption of a bit-string m :
  - Choose r randomly in $\mathbb{Z}_q$
  - Compute $Y = g^r \mod p, Z_1 = Y_1^r \mod p$ and $Z_2 = Y_2^r \mod p$
  - Let $H(Y, Z_1, Z_2) = K||k$ and $c_1 = E_K(m)$
  - Output $c = (Y, E_K(m), MAC_k(c_1))$
- Decryption of $\hat{c} = (\hat{Y}, \hat{c}_1, \hat{c}_2)$
  - Compute $\hat{Z}_1 = \hat{Y}^{x_1} \mod p$ and $\hat{Z}_2 = \hat{Y}^{x_2} \mod p$
  - Let $H(\hat{Y}, \hat{Z}_1, \hat{Z}_2) = \hat{K}||\hat{k}$
  - Output $D_{\hat{K}}(\hat{c}_1)$ if $MAC_{\hat{k}}(\hat{c}_1) = \hat{c}_2$. Otherwise, output “reject"

Theorem

Twin ElGamal PKE is IND-CCA secure if the Computational DH assumption holds in G_q, (E, D) is an IND-CPA symmetric cipher and MAC is unforgeable, in the Random Oracle Model