Cramer-Shoup Encryption

Rongmao Chen
University of Wollongong

August 29, 2014

Literature

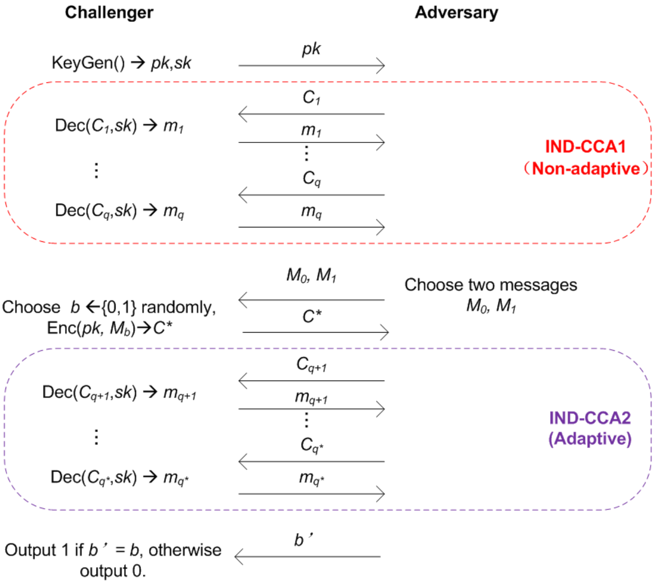
Ronald Cramer and Victor Shoup.
A practical public key cryptosystems provably secure against adaptive chosen ciphertext attack.
In *CRYPTO*, pages 13–25, 1998.

Public Key Encryption

A public key encryption (PKE) scheme consists of the following algorithms,

- **KeyGen:** Taking as input a security parameter 1^λ, return a public/secret key pair (pk, sk).
- **Enc:** Taking as input a plaintext m and the public key pk, return the ciphertext c.
- **Dec:** Taking as input a ciphertext c and the secret key sk, return the plaintext m or $\perp$.

Security Model



Contribution of Cramer-Shoup Encryption

Before the Cramer-Shoup encryption scheme, all the proposed PKE schemes provably secure against adaptive chosen ciphertext attack suffer from either of the following weaknesses.

- Provably secure under standard assumptions but impractical. (none-interactive zero-knowledge proof)
- Practical but provably secure under non-standard assumption. (random oracle)

Contribution of Cramer-Shoup Encryption

Before the Cramer-Shoup encryption scheme, all the proposed PKE schemes provably secure against adaptive chosen ciphertext attack suffer from either of the following weaknesses.

- Provably secure under standard assumptions but impractical. (none-interactive zero-knowledge proof)
- Practical but provably secure under non-standard assumption. (random oracle)

While, the CS scheme is both practical and provably secure under standard assumption.

Let $\mathbb{G}$ be a group of prime order p and $H : \{0,1\}^* \to \mathbb{Z}_p$ be a secure one-way function, $g_1, g_2 \in \mathbb{G}$.

- **KeyGen**: $sk = (\alpha, \beta_1, \beta_2, \gamma_1, \gamma_2) \in \mathbb{Z}_p^6$, $pk = (g_1, g_2, h, u, v) = (g_1,\ g_2,\ g_1^{\alpha},\ g_1^{\beta_1} g_2^{\beta_2},\ g_1^{\gamma_1} g_2^{\gamma_2})$.
- **Enc**$_{pk}(m)$: $r \leftarrow_R \mathbb{Z}_p$, output

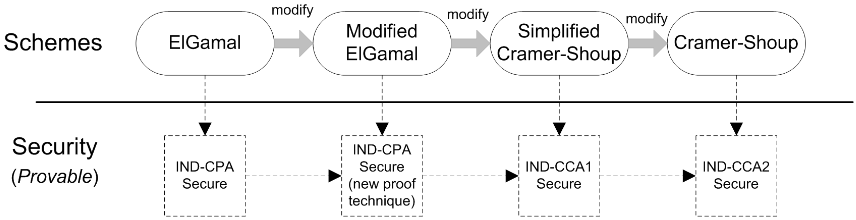
$$CT = < C_1, C_2, C_3, C_4 > = < g_1^r,\ g_2^r,\ h^r m,\ u^r v^{r\theta} >,$$

  where $\theta = H(C_1, C_2, C_3)$.
- **Dec**$_{sk}(C_1, C_2, C_3, C_4)$: If $C_4 = C_1^{\beta_1 + \theta\gamma_1} C_2^{\beta_2 + \theta\gamma_2}$, where $\theta = H(C_1, C_2, C_3)$, output

$$m = C_3 \cdot C_1^{-\alpha},$$

  otherwise output $\perp$.

Schemes to Describe



Reduction Proof–"Guess" Reduction

What is "Guess" Reduction?

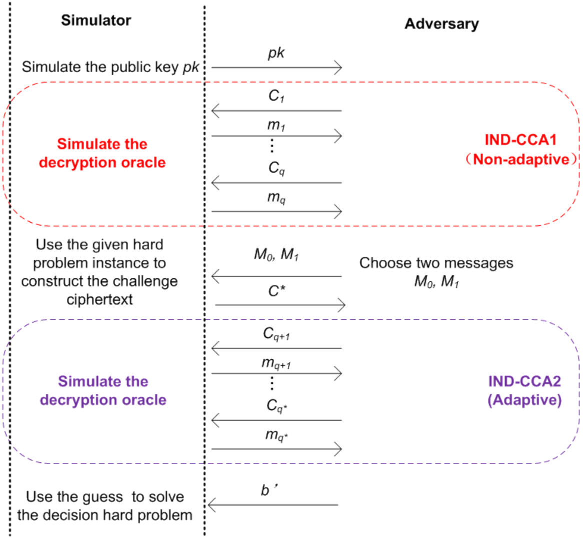
- Solve the hard problem based on the adversary's final guess in the security model;
- Always reduction to decision hard problem, e.g., DDH;
- Sketchy of the reduction proof
  - **Case 1:** The input decision problem is *True*. Prove that the simulation is polynomially indistinguishable from the actual attack;
  - **Case 2:** The input decision problem is *False*. Prove that the challenge ciphertext is "one-time pad" encryption from the view of the adversary.

One-time pad encryption:

Given the ciphertext, any message from the message space has the same probability to be the corresponding plaintext!

It is a challenge to prove the "one-time pad"!!

"Guess" Reduction Map



ElGamal Encryption

Let $\mathbb{G}$ be a group of prime order p, $g \in \mathbb{G}$.

- **KeyGen**: $sk = \alpha \in \mathbb{Z}_p, pk = (g, h)$ where $h = g^\alpha$.
- $\mathbf{Enc}_{pk}(m)$: $r \leftarrow_R \mathbb{Z}_p$, output

$$CT =< C_1, C_2 >=< g^r,\ h^r m > .$$

- $\mathbf{Dec}_{sk}(C_1, C_2)$: Output $m = C_2 \cdot C_1^{-\alpha}$.

ElGamal Encryption

Let $\mathbb{G}$ be a group of prime order p, $g \in \mathbb{G}$.

- **KeyGen**: $sk = \alpha \in \mathbb{Z}_p, pk = (g, h)$ where $h = g^{\alpha}$.
- $\textbf{Enc}_{pk}(m)$: $r \leftarrow_R \mathbb{Z}_p$, output

$$CT = < C_1, C_2 > = < g^r,\ h^r m > .$$

- $\textbf{Dec}_{sk}(C_1, C_2)$: Output $m = C_2 \cdot C_1^{-\alpha}$.

Proof for IND-CPA Security

DDH: Given $< g, g^a, g^b, Z >$, decide $Z \stackrel{?}{=} g^{ab}$.

Suppose $\mathcal{A}$ is an IND-CPA attacker on the ElGamal scheme with advantage ε.

Reduction algorithm $\mathcal{B}(g, g^a, g^b, Z)$

- **KeyGen**: $\mathcal{B}$ gives $\mathcal{A}$ the public key $pk = (g, g^a)$.
- **Challenge**: After $\mathcal{A}$ outputs two messages m_0, m_1, $\mathcal{B}$ chooses $c \leftarrow_R \{0, 1\}$ and outputs

$$CT^* =< C_1^*, C_2^* >=< g^b, Z \cdot m_c > .$$

- **Output**: After $\mathcal{A}$ outputs its guess c' on c, $\mathcal{B}$ outputs 1 if $c' = c$, otherwise outputs 0.

Case 1: $Z = g^{ab}$. The simulation is indistinguishable from the actual attack, that is $P[c' = c | Z = g^{ab}] = \varepsilon$. $\surd$

Case 2: $Z \neq g^{ab}$. As Z is random and independent of $\mathcal{A}$'s view, Z is a perfect one-time pad, that is $P[c' = c | Z \neq g^{ab}] = 1/2. \surd$

Therefore, $\mathcal{B}$ solves the DDH problem with probability,

$$\varepsilon' = \varepsilon - 1/2.$$

Variant of DDH Problem

Given $D =< g_1, g_2, u_1, u_2 >$, if there exist an r that $u_1 = g_1^r$, $u_2 = g_2^r$, then D is a DDH-tuple.

Modified ElGamal Encryption

Let $\mathbb{G}$ be a group of prime order p, $g_1, g_2 \in \mathbb{G}$.

- **KeyGen**: $sk = (\alpha_1, \alpha_2) \in \mathbb{Z}_p^2$, $pk = (g, h)$ where $h = g_1^{\alpha_1} g_2^{\alpha_2}$.
- **Enc**$_{pk}(m)$: $r \leftarrow_R \mathbb{Z}_p$, output

$$CT =< C_1, C_2, C_3 >=< g_1^r, g_2^r, h^r \cdot m > .$$

- **Dec**$_{sk}(C_1, C_2, C_3)$: Output $m = C_3 \cdot C_1^{-\alpha_1} \cdot C_2^{-\alpha_2}$.

Modified ElGamal Encryption

Let $\mathbb{G}$ be a group of prime order p, $g_1, g_2 \in \mathbb{G}$.

- **KeyGen**: $sk = (\alpha_1, \alpha_2) \in \mathbb{Z}_p^2, pk = (g, h)$ where $h = g_1^{\alpha_1} g_2^{\alpha_2}$.
- **Enc**$_{pk}(m)$: $r \leftarrow_R \mathbb{Z}_p$, output

$$CT = < C_1, C_2, C_3 > = < g_1^r, g_2^r, h^r \cdot m > .$$

- **Dec**$_{sk}(C_1, C_2, C_3)$: Output $m = C_3 \cdot C_1^{-\alpha_1} \cdot C_2^{-\alpha_2}$.

IND-CPA Secure?

Given DDH instance $D = < g_1, g_2, u_1, u_2 >$, suppose the challenge ciphertext is

$$CT^* = < C_1^*, C_2^*, C_3^* > = < u_1, u_2, u_1^{\alpha_1} u_2^{\alpha_2} m_b >$$

Let $\log_{g_1}(\cdot) = \log(\cdot)$, suppose that $\log g_2 = w$, then from the public key, we have

$$\log h = \alpha_1 + w\alpha_2 \quad (1)$$

Case 1: D is a DDH-tuple. The simulation is indistinguishable from the actual attack. $\surd$

Modified ElGamal Encryption

Let $\log_{g_1}(\cdot) = \log(\cdot)$, suppose that $\log g_2 = w$, then from the public key, we have

$$\log h = \alpha_1 + w\alpha_2 \qquad (1)$$

Case 1: D is a DDH-tuple. The simulation is indistinguishable from the actual attack. $\surd$

Case 2: D is not a DDH-tuple. Suppose that $u_1 = g_1^{r_1}, u_2 = g_2^{r_2}$, consider the term $u_1^{\alpha} u_2^{\alpha_2}$, we have

$$\log u_1^{\alpha} u_2^{\alpha_2} = r_1\alpha_1 + r_2 w\alpha_2 \qquad (2)$$

As equation (2) is linearly independent from equation (1), $u_1^{\alpha_1} u_2^{\alpha_2}$ is independent of $\mathcal{A}$'s view, which follows that C_3^* is one-time pad encryption. $\surd$

Modified ElGamal Encryption

Let $\log_{g_1}(\cdot) = \log(\cdot)$, suppose that $\log g_2 = w$, then from the public key, we have

$$\log h = \alpha_1 + w\alpha_2 \qquad (1)$$

Case 1: D is a DDH-tuple. The simulation is indistinguishable from the actual attack. $\surd$

Case 2: D is not a DDH-tuple. Suppose that $u_1 = g_1^{r_1}, u_2 = g_2^{r_2}$, consider the term $u_1^{\alpha} u_2^{\alpha_2}$, we have

$$\log u_1^{\alpha} u_2^{\alpha_2} = r_1\alpha_1 + r_2 w\alpha_2 \qquad (2)$$

As equation (2) is linearly independent from equation (1), $u_1^{\alpha_1} u_2^{\alpha_2}$ is independent of $\mathcal{A}$'s view, which follows that C_3^* is one-time pad encryption. $\surd$

IND-CPA Secure!

Modified ElGamal Encryption

Let $\log_{g_1}(\cdot) = \log(\cdot)$, suppose that $\log g_2 = w$, then from the public key, we have

$$\log h = \alpha_1 + w\alpha_2 \tag{1}$$

Case 1: D is a DDH-tuple. The simulation is indistinguishable from the actual attack. $\surd$

Case 2: D is not a DDH-tuple. Suppose that $u_1 = g_1^{r_1}, u_2 = g_2^{r_2}$, consider the term $u_1^{\alpha} u_2^{\alpha_2}$, we have

$$\log u_1^{\alpha} u_2^{\alpha_2} = r_1\alpha_1 + r_2 w\alpha_2 \tag{2}$$

As equation (2) is linearly independent from equation (1), $u_1^{\alpha_1} u_2^{\alpha_2}$ is independent of $\mathcal{A}$'s view, which follows that C_3^* is one-time pad encryption. $\surd$

IND-CPA Secure!

IND-CCA1 Secure?

IND-CCA1 Secure?

Suppose that $\mathcal{A}$ submit an invalid ciphertext to the decryption oracle , say $< C_1', C_2', C_3' >$, where $C_1' = g_1^{r_1'}, C_2' = g_2^{r_2'}$ and $r_1' \neq r_2'$. Using the decryption result m', $\mathcal{A}$ has the following info,

$$\log C_3'/m' = r_1'\alpha_1 + r_2' w \alpha_2 \qquad (3)$$

Since equations (1), (3) are linearly independent, $\mathcal{A}$ can solve the linear equations to get the value of α_1, α_2, i.e., the secret key.

IND-CCA1 Secure?

Suppose that $\mathcal{A}$ submit an invalid ciphertext to the decryption oracle , say $< C_1', C_2', C_3' >$, where $C_1' = g_1^{r_1'}, C_2' = g_2^{r_2'}$ and $r_1' \neq r_2'$.

Using the decryption result m', $\mathcal{A}$ has the following info,

$$\log C_3'/m' = r_1'\alpha_1 + r_2' w \alpha_2 \quad (3)$$

Since equations (1), (3) are linearly independent, $\mathcal{A}$ can solve the linear equations to get the value of α_1, α_2, i.e., the secret key.

Case 2 can not be proved!

IND-CCA1 Secure?

Suppose that $\mathcal{A}$ submit an invalid ciphertext to the decryption oracle , say $< C_1', C_2', C_3' >$, where $C_1' = g_1^{r_1'}, C_2' = g_2^{r_2'}$ and $r_1' \neq r_2'$.

Using the decryption result m', $\mathcal{A}$ has the following info,

$$\log C_3'/m' = r_1'\alpha_1 + r_2' w\alpha_2 \tag{3}$$

Since equations (1), (3) are linearly independent, $\mathcal{A}$ can solve the linear equations to get the value of α_1, α_2, i.e., the secret key.

Case 2 can not be proved!

Fail to prove IND-CCA1 security!

IND-CCA1 Secure?

Suppose that $\mathcal{A}$ submit an invalid ciphertext to the decryption oracle , say $< C_1', C_2', C_3' >$, where $C_1' = g_1^{r_1'}, C_2' = g_2^{r_2'}$ and $r_1' \neq r_2'$. Using the decryption result m', $\mathcal{A}$ has the following info,

$$\log C_3'/m' = r_1'\alpha_1 + r_2' w \alpha_2 \quad (3)$$

Since equations (1), (3) are linearly independent, $\mathcal{A}$ can solve the linear equations to get the value of α_1, α_2, i.e., the secret key.

Case 2 can not be proved!

Fail to prove IND-CCA1 security!

Solution: Check the validity of the ciphertext before decryption⇒ Proving consistency of exponentiations, i.e., ensure that,

$$\log_{g_1} C_1' = \log_{g_2} C_2'?$$

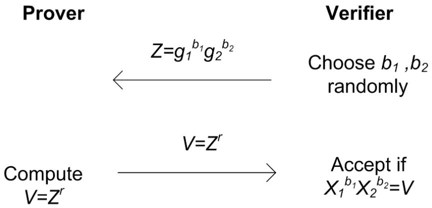
Proving Consistency of Exponentiations

Q: Given g_1, g_2, X_1, X_2, prove that there is an r where $X_1 = g_1^r$, $X_2 = g_2^r$.

Proving consistency of exponentiations

Proving Consistency of Exponentiations

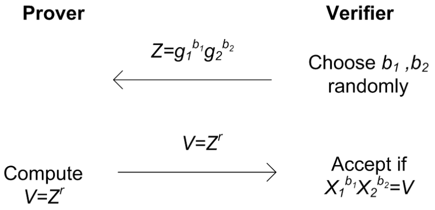
Q: Given g_1, g_2, X_1, X_2, prove that there is an r where $X_1 = g_1^r$, $X_2 = g_2^r$.



Proving consistency of exponentiations

Proving Consistency of Exponentiations

Q: Given g_1, g_2, X_1, X_2, prove that there is an r where $X_1 = g_1^r$, $X_2 = g_2^r$.



Soundness: if $X_1 = g_1^{r_1}, X_2 = g_2^{r_2} = g_2^{r_1+\Delta r}$, then

$$X_1^{b_1} X_2^{b_2} = g_1^{r_1 b_1} g_2^{(r_1+\Delta r) b_2} = g_1^{r_1 b_1} g_2^{r_1 b_2} g_2^{\Delta r b_2} = Z^{r_1} (g_2^{\Delta r})^{b_2}$$

Independent of the prover's view!

Simplified Cramer-Shoup Encryption

Let $\mathbb{G}$ be a group of prime order p, $g_1, g_2 \in \mathbb{G}$.

- **KeyGen**: $sk = (\alpha_1, \alpha_2, \beta_1, \beta_2) \in \mathbb{Z}_p^4$, $pk = (g, h, u)$ where $h = g_1^{\alpha_1} g_2^{\alpha_2}$, $u = g_1^{\beta_1} g_2^{\beta_2}$.
- **Enc**$_{pk}(m)$: $r \leftarrow_R \mathbb{Z}_p$, output

$$CT =< C_1, C_2, C_3, C_4 >=< g_1^r, g_2^r, h^r \cdot m, u^r > .$$

- **Dec**$_{sk}(C_1, C_2, C_3, C_4)$: If $C_4 = C_1^{\beta_1} C_2^{\beta_2}$, output

$$m = C_3 \cdot C_1^{-\alpha_1} \cdot C_2^{-\alpha_2},$$

  otherwise output $\perp$.

IND-CCA1 Secure?

From the public key u, $\mathcal{A}$ gets the following info,

$$\log u = \beta_1 + w\beta_2 \tag{4}$$

For a query $< C_1', C_2', C_3', C_4' >$, $C_1' = g_1^{r_1'}$, $C_2' = g_2^{r_2'}$, $r_1' \neq r_2'$. If it is accepted, then $C_4' = C_1'^{\beta_1} C_2'^{\beta_2}$, i.e, the following equation,

$$\log C_4' = r_1'\beta_1 + r_2' w\beta_2 \tag{5}$$

Since equations (4), (5) are linearly independent, this happens with only negligible probability.

Simplified Cramer-Shoup Encryption

IND-CCA1 Secure?

From the public key u, $\mathcal{A}$ gets the following info,

$$\log u = \beta_1 + w\beta_2 \tag{4}$$

For a query $< C_1', C_2', C_3', C_4' >$, $C_1' = g_1^{r_1'}$, $C_2' = g_2^{r_2'}$, $r_1' \neq r_2'$. If it is accepted, then $C_4' = C_1'^{\beta_1} C_2'^{\beta_2}$, i.e, the following equation,

$$\log C_4' = r_1'\beta_1 + r_2' w\beta_2 \tag{5}$$

Since equations (4), (5) are linearly independent, this happens with only negligible probability.

Validity checking works! $\Rightarrow$ **Case 2** can be proved! $\surd$

IND-CCA1 Secure?

From the public key u, $\mathcal{A}$ gets the following info,

$$\log u = \beta_1 + w\beta_2 \tag{4}$$

For a query $< C_1', C_2', C_3', C_4' >$, $C_1' = g_1^{r_1'}, C_2' = g_2^{r_2'}, r_1' \neq r_2'$. If it is accepted, then $C_4' = C_1'^{\beta_1} C_2'^{\beta_2}$, i.e, the following equation,

$$\log C_4' = r_1'\beta_1 + r_2' w\beta_2 \tag{5}$$

Since equations (4), (5) are linearly independent, this happens with only negligible probability.

Validity checking works! $\Rightarrow$**Case 2** can be proved!$\surd$

IND-CCA1 secure!

IND-CCA1 Secure?

From the public key u, $\mathcal{A}$ gets the following info,

$$\log u = \beta_1 + w\beta_2 \tag{4}$$

For a query $< C_1', C_2', C_3', C_4' >$, $C_1' = g_1^{r_1'}$, $C_2' = g_2^{r_2'}$, $r_1' \neq r_2'$. If it is accepted, then $C_4' = C_1'^{\beta_1} C_2'^{\beta_2}$, i.e, the following equation,

$$\log C_4' = r_1'\beta_1 + r_2' w\beta_2 \tag{5}$$

Since equations (4), (5) are linearly independent, this happens with only negligible probability.

Validity checking works! $\Rightarrow$ **Case 2** can be proved! $\surd$

IND-CCA1 secure!

IND-CCA2 secure?

IND-CCA2 Secure?

Suppose that the challenge ciphertext is as follows,

$$CT^* = < C_1^*, C_2^*, C_3^*, C_4^* > = < u_1,\ u_2,\ u_1^{\alpha_1} u_2^{\alpha_2} m_b,\ u_1^{\beta_1} u_2^{\beta_2} >$$

There are two aspects need to be considered.

- **Malleability.** $\mathcal{A}$ chooses Δm randomly and submits the follow ciphertext to the decryption oracle.

  $$CT = < C_1^*, C_2^*, C_3^* \cdot \Delta m, C_4^* > = < u_1, u_2, u_1^{\alpha_1} u_2^{\alpha_2} m_b \cdot \Delta m, u_1^{\beta_1} u_2^{\beta_2} >$$

  Since $CT \neq CT^*$ and is a valid ciphertext, the decryption oracle returns $m' = m_b \cdot \Delta m$ to $\mathcal{A}$. Thus $\mathcal{A}$ can compute $m_b = m' / \Delta m$ and output its guess correctly regardless of tuple D.

IND-CCA2 Secure?

Suppose that the challenge ciphertext is as follows,

$$CT^* =< C_1^*, C_2^*, C_3^*, C_4^* >=< u_1,\ u_2,\ u_1^{\alpha_1} u_2^{\alpha_2} m_b,\ u_1^{\beta_1} u_2^{\beta_2} >$$

There are two aspects need to be considered.

- **Malleability.** $\mathcal{A}$ chooses Δm randomly and submits the follow ciphertext to the decryption oracle.

  $$CT =< C_1^*, C_2^*, C_3^* \cdot \Delta m, C_4^* >=< u_1, u_2, u_1^{\alpha_1} u_2^{\alpha_2} m_b \cdot \Delta m, u_1^{\beta_1} u_2^{\beta_2} >$$

  Since $CT \neq CT^*$ and is a valid ciphertext, the decryption oracle returns $m' = m_b \cdot \Delta m$ to $\mathcal{A}$. Thus $\mathcal{A}$ can compute $m_b = m' / \Delta m$ and output its guess correctly regardless of tuple D.

IND-CCA2 insecure!

IND-CCA2 Secure?

Suppose that the challenge ciphertext is as follows,

$$CT^* =< C_1^*, C_2^*, C_3^*, C_4^* >=< u_1,\ u_2,\ u_1^{\alpha_1} u_2^{\alpha_2} m_b,\ u_1^{\beta_1} u_2^{\beta_2} >$$

There are two aspects need to be considered.

- **Malleability.** $\mathcal{A}$ chooses Δm randomly and submits the follow ciphertext to the decryption oracle.

  $$CT =< C_1^*, C_2^*, C_3^* \cdot \Delta m, C_4^* >=< u_1, u_2, u_1^{\alpha_1} u_2^{\alpha_2} m_b \cdot \Delta m, u_1^{\beta_1} u_2^{\beta_2} >$$

  Since $CT \neq CT^*$ and is a valid ciphertext, the decryption oracle returns $m' = m_b \cdot \Delta m$ to $\mathcal{A}$. Thus $\mathcal{A}$ can compute $m_b = m' / \Delta m$ and output its guess correctly regardless of tuple D.

IND-CCA2 insecure!

Solution: Use the message info for validity checking !

- **Validity Checking Failure**. Suppose that D is not a DDH tuple $(u_1 = g_1^{r_1}, u_2 = g_2^{r_2}, r_1 \neq r_2)$. Based on the challenge ciphertext, the (powerful) adversary $\mathcal{A}$ can get the following info,

$$\log C_4^* = r_1\beta_1 + r_2 w\beta_2 \tag{6}$$

Since equations (4),(6) are linearly independent, $\mathcal{A}$ can solve the linear equations to get the value of β_1, β_2. It follows that the ciphertext *validity checking would be a failure*.

- **Validity Checking Failure**. Suppose that D is not a DDH tuple $(u_1 = g_1^{r_1}, u_2 = g_2^{r_2}, r_1 \neq r_2)$. Based on the challenge ciphertext, the (powerful) adversary $\mathcal{A}$ can get the following info,

$$\log C_4^* = r_1\beta_1 + r_2 w\beta_2 \tag{6}$$

Since equations (4),(6) are linearly independent, $\mathcal{A}$ can solve the linear equations to get the value of β_1, β_2. It follows that the ciphertext *validity checking would be a failure*.

Validity checking Fails! $\Rightarrow$ **Case 2** cannot be proved!

- **Validity Checking Failure**. Suppose that D is not a DDH tuple ($u_1 = g_1^{r_1}, u_2 = g_2^{r_2}, r_1 \neq r_2$). Based on the challenge ciphertext, the (powerful) adversary $\mathcal{A}$ can get the following info,

$$\log C_4^* = r_1\beta_1 + r_2 w\beta_2 \tag{6}$$

Since equations (4),(6) are linearly independent, $\mathcal{A}$ can solve the linear equations to get the value of β_1, β_2. It follows that the ciphertext *validity checking would be a failure*.

Validity checking Fails! $\Rightarrow$ **Case 2** cannot be proved!

Still fail to prove IND-CCA2 security!

- **Validity Checking Failure**. Suppose that D is not a DDH tuple ($u_1 = g_1^{r_1}, u_2 = g_2^{r_2}, r_1 \neq r_2$). Based on the challenge ciphertext, the (powerful) adversary $\mathcal{A}$ can get the following info,

$$\log C_4^* = r_1\beta_1 + r_2 w\beta_2 \quad (6)$$

Since equations (4),(6) are linearly independent, $\mathcal{A}$ can solve the linear equations to get the value of β_1, β_2. It follows that the ciphertext *validity checking would be a failure*.

Validity checking Fails! $\Rightarrow$ **Case 2** cannot be proved!

Still fail to prove IND-CCA2 security!

Solution: Use more random augments for validity checking!

Cramer-Shoup Encryption

Let $\mathbb{G}$ be a group of prime order p and $H : \{0,1\}^* \to \mathbb{Z}_p$ be a secure one-way function, $g_1, g_2 \in \mathbb{G}$.

- **KeyGen**: $sk = (\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma_1, \gamma_2) \in \mathbb{Z}_p^6$, $pk = (g, h, u, v)$ where $h = g_1^{\alpha_1} g_2^{\alpha_2}$, $u = g_1^{\beta_1} g_2^{\beta_2}$, $v = g_1^{\gamma_1} g_2^{\gamma_2}$.
- **Enc**$_{pk}(m)$: $r \leftarrow_R \mathbb{Z}_p$, output

$$CT = < C_1, C_2, C_3, C_4 > = < g_1^r, g_2^r, h^r m, u^r v^{r\theta} >,$$

  where $\theta = H(C_1, C_2, C_3)$.
- **Dec**$_{sk}(C_1, C_2, C_3, C_4)$: If $C_4 = C_1^{\beta_1 + \theta\gamma_1} C_2^{\beta_2 + \theta\gamma_2}$, where $\theta = H(C_1, C_2, C_3)$, output

$$m = C_3 \cdot C_1^{-\alpha_1} \cdot C_2^{-\alpha_2},$$

  otherwise output $\perp$.

IND-CCA2 Secure?

From the public key v, $\mathcal{A}$ can get the following info,

$$\log v = \gamma_1 + w\gamma_2 \tag{7}$$

Suppose that D is not a DDH tuple ($u_1 = g_1^{r_1}, u_2 = g_2^{r_2}, r_1 \neq r_2$), then the challenge ciphertext is as follows,

$$CT^* =< C_1^*, C_2^*, C_3^*, C_4^* >=< u_1, u_2, u_1^{\alpha_1} u_2^{\alpha_2} m_b, u_1^{\beta_1} u_2^{\beta_2} u_1^{\gamma_1\theta^*} u_2^{\gamma_2\theta^*} >$$

where $\theta^* = H(C_1^*, C_2^*, C_3^*)$. Therefore , $\mathcal{A}$ can get the following info,

$$\log C_4^* = r_1\beta_1 + r_2 w\beta_2 + r_1\gamma_1\theta^* + r_2 w\gamma_2\theta^* \tag{8}$$

If $\mathcal{A}$ queries an invalid ciphertext to the decryption oracle, say $< C_1', C_2', C_3', C_4' >$, where $C_1' = g_1^{r_1'}, C_2' = g_2^{r_2'}$ and $r_1' \neq r_2'$. As for this decryption query, we should consider the followings.

- If $< C_1', C_2', C_3' > = < C_1^*, C_2^*, C_3^* >, C_4' \neq C_4^*$. This query will always be rejected.
- If $< C_1', C_2', C_3' > \neq < C_1^*, C_2^*, C_3^* >, C_4' = C_4^*$. Since H is collision-resistant and $\mathcal{A}$ runs in polynomial time,this happens with only negligible probability.
- If $H(C_1', C_2', C_3') \neq H(C_1^*, C_2^*, C_3^*)$. If the ciphertext is accepted by the simulator, it should satisfy the following equation,

$$\log C_4' = r_1'\beta_1 + r_2' w\beta_2 + r_1'\gamma_1\theta' + r_2' w\gamma_2\theta' \qquad (9)$$

where $\theta' = H(C_1', C_2', C_3')$. Since equations (4), (7), (8), (9) are linearly independent, this happens only with negligible probability.

- If $< C_1', C_2', C_3' >=< C_1^*, C_2^*, C_3^* >, C_4' \neq C_4^*$. This query will always be rejected.
- If $< C_1', C_2', C_3' >\neq< C_1^*, C_2^*, C_3^* >, C_4' = C_4^*$. Since H is collision-resistant and $\mathcal{A}$ runs in polynomial time,this happens with only negligible probability.
- If $H(C_1', C_2', C_3') \neq H(C_1^*, C_2^*, C_3^*)$. If the ciphertext is accepted by the simulator, it should satisfy the following equation,

$$\log C_4' = r_1'\beta_1 + r_2'w\beta_2 + r_1'\gamma_1\theta' + r_2'w\gamma_2\theta' \tag{9}$$

where $\theta' = H(C_1', C_2', C_3')$. Since equations (4), (7), (8), (9) are linearly independent, this happens only with negligible probability.

Validity checking works! $\Rightarrow$**Case 2** can be proved!$\surd$

- If $< C_1', C_2', C_3' > = < C_1^*, C_2^*, C_3^* >, C_4' \neq C_4^*$. This query will always be rejected.
- If $< C_1', C_2', C_3' > \neq < C_1^*, C_2^*, C_3^* >, C_4' = C_4^*$. Since H is collision-resistant and $\mathcal{A}$ runs in polynomial time,this happens with only negligible probability.
- If $H(C_1', C_2', C_3') \neq H(C_1^*, C_2^*, C_3^*)$. If the ciphertext is accepted by the simulator, it should satisfy the following equation,

$$\log C_4' = r_1'\beta_1 + r_2'w\beta_2 + r_1'\gamma_1\theta' + r_2'w\gamma_2\theta' \tag{9}$$

  where $\theta' = H(C_1', C_2', C_3')$. Since equations (4), (7), (8), (9) are linearly independent, this happens only with negligible probability.

Validity checking works! $\Rightarrow$ **Case 2** can be proved! $\surd$

IND-CCA2 secure!

What can we learn from CS scheme?

- Some schemes seem to be secure without attacks, but they cannot be proved. We must change schemes to make them provably secure.
- Use "guarded" decryption, i.e., checking the validity of ciphertext before or after decryption to remove the scheme's property of malleability to achieve IND-CCA2 security.
- To construct a practical PKE scheme that is IND-CCA2 secure, "guess" reduction is a useful technique to proof its security under standard assumption.
  (how to prove the **Case 2** is the key part, i.e, analysis the relationship between the challenge ciphertext and all the information that adversary can get.)
- Adversary sometimes is suppose to be computation-unlimited to make the scheme security provable.
  (to bound the advantage of the adversary)

Thank you

Thank you

Any questions?