Introduction to Cryptology

10.1 & 10.2 - Cramer-Shoup Encryption Scheme, and Hash Functions

Federico Pintore

Mathematical Institute, University of Oxford (UK)

UNIVERSITY OF OXFORD

Michaelmas term 2020

Cramer-Shoup Encryption Scheme

Proposed by Ronald Cramer and Victor Shoup in 1998. It is based on the ElGamal Encryption Scheme.

It was the first efficient public-key encryption scheme proven to be CCA-secure in the standard model.

Its CCA-security relies on the hardness of the DDH problem.

Cramer-Shoup Encryption Scheme

It is relative to a group generation algorithm $\mathcal{G}$ that, on input a security parameter n, returns:

- a description of a cyclic group $\mathbb{G}$ having prime order q, where $||q|| = \lfloor \log_2 q \rfloor + 1 = n$;
- a couple of generators g_1, g_2 for $\mathbb{G}$.

Cramer-Shoup Encryption Scheme

It is relative to a group generation algorithm $\mathcal{G}$ that, on input a security parameter n, returns:

- a description of a cyclic group $\mathbb{G}$ having prime order q, where $||q|| = \lfloor \log_2 q \rfloor + 1 = n$;
- a couple of generators g_1, g_2 for $\mathbb{G}$.

The Cramer-Shoup encryption scheme relative to $\mathcal{G}$

$$CS = (\mathrm{KeyGen}, \mathrm{Enc}, \mathrm{Dec})$$

is defined as follows.

Cramer-Shoup Encryption Scheme

- $(\mathrm{PK}, \mathrm{SK}) \leftarrow \mathrm{KeyGen}(n)$: it runs $\mathcal{G}$ on input a security parameter n, obtaining a group $\mathbb{G}$, its order q, and a couple of generators g_1, g_2 for $\mathbb{G}$.

  Then, it specifies a collision-resistant hash function $H : \{0,1\}^* \to \{1, \ldots, q\}$, picks uniform x_1, x_2, y_1, y_2, w_1, $w_2 \in \{1, \ldots, q\}$ and computes:

  - $c := g_1^{x_1} g_2^{x_2}$;
  - $d := g_1^{y_1} g_2^{y_2}$;
  - $h := g_1^{w_1} g_2^{w_2}$.

  The public key is $\mathrm{PK} = (\mathbb{G}, q, g_1, g_2, c, d, h, H)$.

  The secret key is $\mathrm{SK} = (x_1, x_2, y_1, y_2, w_1, w_2)$.

Cramer-Shoup Encryption Scheme

- $CT \leftarrow \mathrm{Enc}(\mathrm{PK}, m \in \mathbb{G})$: on input a public key PK and a message m, it chooses a uniform $k \in \mathbb{Z}_q$, and computes:
  - $u_1 = g_1^k, u_2 = g_2^k$;
  - $e = h^k m$;
  - $\alpha = H(u_1, u_2, e)$;
  - $v = c^k d^{k\alpha}$.

  The ciphertext CT is (u_1, u_2, e, v).

Cramer-Shoup Encryption Scheme

- $m \leftarrow \mathrm{Dec}(CT, \mathrm{SK})$: on input a ciphertext $CT = (u_1, u_2, e, v)$ and a secret key $\mathrm{SK} = (x_1, x_2, y_1, y_2, z_1, z_2)$, it computes $\alpha = H(u_1, u_2, e)$.

  If $u_1^{x_1} u_2^{x_2} (u_1^{y_1} u_2^{y_2})^{\alpha} \neq v$, it outputs $\perp$.

  Otherwise it outputs $m = e/(u_1^{w_1} u_2^{w_2})$

Correctness: $e/(u_1^{w_1} u_2^{w_2}) = h^k m / g_1^{kw_1} g_2^{kw_2} = h^k m / h^k = m$.

Cramer-Shoup: Security Proof

Proof.

Let $\mathcal{A}$ be a PPT adversary in the experiment $\mathrm{PubK}^{\mathrm{cca}}_{\mathcal{A},\mathcal{CS}}$.

$\mathcal{A}$ is exploited, as a subroutine, to construct a distinguisher D for the DDH problem relative to $\mathcal{G}$.

Cramer-Shoup: Security Proof

Proof.

Let $\mathcal{A}$ be a PPT adversary in the experiment $\mathrm{PubK}^{\mathrm{cca}}_{\mathcal{A},CS}$.

$\mathcal{A}$ is exploited, as a subroutine, to construct a distinguisher D for the DDH problem relative to $\mathcal{G}$.

D receives $(\mathbb{G}, q, g_1, \tilde{g}_2, g_3, g_4)$, picks uniform x_1, x_2, y_1, y_2, $w_1, w_2 \in \{1, \ldots, q\}$ and sets

$$\mathrm{PK} := (\mathbb{G}, q, g_1, \tilde{g}_2, c := g_1^{x_1}\tilde{g}_2^{x_2}, d := g_1^{y_1}\tilde{g}_2^{y_2}, h := g_1^{w_1}\tilde{g}_2^{w_2}, H).$$

PK is sent to $\mathcal{A}$.

Cramer-Shoup: Security Proof

Decryption queries:

On input $(u_1, u_2, e, v) \in \mathbb{G}^4$, D computes $\alpha = H(u_1, u_2, e)$. If

$$u_1^{x_1+\alpha y_1} u_2^{x_2+\alpha y_2} \neq v$$

it outputs $\perp$, otherwise it outputs

$$m' = \frac{e}{u_1^{w_1} u_2^{w_2}}.$$

Cramer-Shoup: Security Proof

D receives (m_0, m_1) from $\mathcal{A}$, picks a uniform bit $b \in \{0,1\}$ and computes

- $e^* = g_3^{w_1} g_4^{w_2} m_b$,
- $\alpha^* = H(g_3, g_4, e^*)$,
- $CT^* = (g_3, g_4, e^*, v^* := g_3^{x_1+\alpha^* y_1} g_4^{x_2+\alpha^* y_2})$.

CT^* is sent to $\mathcal{A}$, who has still access to the decryption oracle.

Cramer-Shoup: Security Proof

D receives (m_0, m_1) from $\mathcal{A}$, picks a uniform bit $b \in \{0,1\}$ and computes

- $e^* = g_3^{w_1} g_4^{w_2} m_b$,
- $\alpha^* = H(g_3, g_4, e^*)$,
- $CT^* = (g_3, g_4, e^*, v^* := g_3^{x_1+\alpha^* y_1} g_4^{x_2+\alpha^* y_2})$.

CT^* is sent to $\mathcal{A}$, who has still access to the decryption oracle.

When D receives $\mathcal{A}$'s guess b', it returns 1 if $b' = b$, 0 otherwise.

Cramer-Shoup: Security Proof

Fact 1: from the hardness of the DDH problem, it follows that

$$\left| \Pr(\mathrm{D} = 1 | \mathrm{DH}) - \Pr(\mathrm{D} = 1 | \mathrm{Random}) \right| \leq \mathrm{negl}_1(n).$$

Fact 2:

$$\Pr(\mathrm{D} = 1 | \mathrm{DH}) = \Pr(\mathrm{PubK}^{\mathrm{cca}}_{\mathcal{A},CS}(n) = 1) + \mathrm{negl}_2(n).$$

Fact 3:

$$\left| \Pr(\mathrm{D} = 1 | \mathrm{Random}) \right| \leq \frac{1}{2} + \mathrm{negl}_3(n).$$

Combining the three facts, the proof follows.

Cramer-Shoup: Security Proof

Proof of Fact 2:

Let I be the event $\tilde{g_2} \in \{1, g_1\}$. Then $\Pr(I|\mathrm{DH}) = 2/q$.

Using the conditional probability and the union formula we obtain: $\Pr(\mathrm{D} = 1|\mathrm{DH}) = \Pr(\mathrm{D} = 1|\mathrm{DH} \cap \bar{I}) + \mathrm{negl}_2(n)$.

Cramer-Shoup: Security Proof

Proof of Fact 2:

Let I be the event $\tilde{g_2} \in \{1, g_1\}$. Then $\Pr(I|\mathrm{DH}) = 2/q$.

Using the conditional probability and the union formula we obtain: $\Pr(\mathrm{D} = 1|\mathrm{DH}) = \Pr(\mathrm{D} = 1|\mathrm{DH} \cap \bar{I}) + \mathrm{negl}_2(n)$.

When D gets a DH tuple with $\tilde{g_2} \notin \{1, g_1\}$, then $\tilde{g}_2$ is a second generator and there exists k s.t.:

$$(g_1, \tilde{g}_2, g_3 = g_1^k, g_4 = \tilde{g}_2^k).$$

In this case, $\mathcal{A}$'s view is distributed exactly as in the game $\mathrm{PubK}^{\mathrm{cca}}_{\mathcal{A},CS}(n)$, and hence:

$$\Pr(\mathrm{D} = 1|\mathrm{DH}) = \Pr(\mathrm{PubK}^{\mathrm{cca}}_{\mathcal{A},CS}(n) = 1) + \mathrm{negl}_2(n).$$

Cramer-Shoup: Security Proof

Proof of Fact 3: (a bit long...)

General idea: even if $\mathcal{A}$ can compute discrete logarithms we have

$$\Pr(\mathrm{D} = 1 | \mathrm{Random}) \leq \frac{1}{2} + \mathrm{negl}(n)'$$

provided $\mathcal{A}$ can make polynomially-many decryption queries.

Cramer-Shoup: Security Proof

Proof of Fact 3: (a bit long...)

General idea: even if $\mathcal{A}$ can compute discrete logarithms we have

$$\Pr(\mathrm{D} = 1 | \mathrm{Random}) \leq \frac{1}{2} + \mathrm{negl}(n)'$$

provided $\mathcal{A}$ can make polynomially-many decryption queries.

When D gets a random tuple, it is of the form

$$(g_1, \tilde{g}_2 = g_1^r, g_3 = g_1^k, g_4 = \tilde{g}_2^{r'})$$

where $r, k, r' \in \{1, \ldots, q\}$. We can assume $r \neq 0$ and $k \neq r'$.

Cramer-Shoup: Security Proof

What does $\mathcal{A}$ learn about w_1, w_2?

Cramer-Shoup: Security Proof

What does $\mathcal{A}$ learn about w_1, w_2?

From the public key PK, $\mathcal{A}$ learns

$$\log_{g_1} h = w_1 + rw_2. \tag{1}$$

Cramer-Shoup: Security Proof

Decryption queries

Consider a decryption query $CT = (u_1, u_2, e, v)$ made by $\mathcal{A}$.

We say that CT is

- illegal if $\log_{g_1} u_1 \neq \log_{\tilde{g}_2} u_2$;
- legal otherwise.

Cramer-Shoup: Security Proof

Decryption queries

Consider a decryption query $CT = (u_1, u_2, e, v)$ made by $\mathcal{A}$.

We say that CT is

- illegal if $\log_{g_1} u_1 \neq \log_{\tilde{g}_2} u_2$;
- legal otherwise.

We will prove that

1. $\mathcal{A}$ does not learn additional information about w_1 and w_2 from legal ciphertexts and from illegal ciphertext for which D returns a message;
2. the probability that D decrypts illegal ciphertexts is negligibly low.

Cramer-Shoup: Security Proof

Assume the validity of the above two points and consider an arbitrary $\mu \in \mathbb{G}$.

The only value in CT^* which directly depends on m_b is $e^* = g_3^{w_1} g_4^{w_2} m_b$.

Suppose $\mu = g_3^{w_1} g_4^{w_2}$. Then:

$$\log_{g_1} \mu = kw_1 + rr'w_2 \tag{2}$$

Cramer-Shoup: Security Proof

Assume the validity of the above two points and consider an arbitrary $\mu \in \mathbb{G}$.

The only value in CT^* which directly depends on m_b is $e^* = g_3^{w_1} g_4^{w_2} m_b$.

Suppose $\mu = g_3^{w_1} g_4^{w_2}$. Then:

$$\log_{g_1} \mu = kw_1 + rr'w_2 \tag{2}$$

Equations (1) and (2) form a system of linear equations in w_1 and w_2 (over $\mathbb{Z}_q$) with matrix of coefficients equal to

$$B = \begin{pmatrix} 1 & r \\ k & rr' \end{pmatrix}$$

which is non singular since $r \neq 0$ and $k \neq r'$.

Cramer-Shoup: Security Proof

Each $\mu \in \mathbb{G}$ is a possible value for $g_3^{w_1} g_4^{w_2}$.

Therefore, the adversary $\mathcal{A}$ cannot predict the value of $g_3^{w_1} g_4^{w_2}$ with probability better than $1/q$.

Since $g_3^{w_1} g_4^{w_2}$ is uniformly distributed in $\mathbb{G}$ from $\mathcal{A}$'s point of view, also $g_3^{w_1} g_4^{w_2} m_b$ is uniformly distributed. Thus $\mathcal{A}$ has no information about m_b.

Cramer-Shoup: Security Proof

1. When $\log_{g_1} u_1 = \log_{\tilde{g}_2} u_2 = r''$, then $\mathcal{A}$ learns from the decrypted message m' that

$$\log_{g_1} m' = \log_{g_1} e - r''w_1 - r''rw_2 \tag{3}$$

But equation (3) is linearly dependent with equation (1), so no extra information about w_1, w_2 in this case.

Cramer-Shoup: Security Proof

1. When $\log_{g_1} u_1 = \log_{\tilde{g}_2} u_2 = r''$, then $\mathcal{A}$ learns from the decrypted message m' that

$$\log_{g_1} m' = \log_{g_1} e - r'' w_1 - r'' r w_2 \tag{3}$$

But equation (3) is linearly dependent with equation (1), so no extra information about w_1, w_2 in this case.

When D returns $\perp$, it means that

$$v \neq u_1^{x_1 + y_1 H(u_1, u_2, e)} u_2^{x_2 + y_2 H(u_1, u_2, e)}.$$

Since w_1, w_2 are not involved in this check, also in this case no information about them is leaked.

Cramer-Shoup: Security Proof

2 We consider two phases: before the challenge ciphertext is sent, and after.

Cramer-Shoup: Security Proof

2 We consider two phases: before the challenge ciphertext is sent, and after.

<u>Before the challenge ciphertext is sent</u>

From the public key PK, $\mathcal{A}$ learns the following about x_1, x_2, y_1, y_2:

$$\log_{g_1} c = x_1 + rx_2 \tag{4}$$

$$\log_{g_1} d = y_1 + ry_2 \tag{5}$$

From $\mathcal{A}$'s point of view, there are q^2 possibilities for x_1, x_2, y_1, y_2.

Cramer-Shoup: Security Proof

Consider an arbitrary $\mu \in \mathbb{G}$, and suppose $\mu = u_1^{x_1+\alpha y_1} u_2^{x_2+\alpha y_2}$. Then we have:

$$\log_{g_1} \mu = r''(x_1 + \alpha y_1) + rr'''(x_2 + \alpha y_2) \tag{6}$$

Cramer-Shoup: Security Proof

Consider an arbitrary $\mu \in \mathbb{G}$, and suppose $\mu = u_1^{x_1+\alpha y_1} u_2^{x_2+\alpha y_2}$. Then we have:

$$\log_{g_1} \mu = r''(x_1 + \alpha y_1) + rr'''(x_2 + \alpha y_2) \tag{6}$$

Equations (4), (5) and (6) form a system of linear equations in x_1, x_2, y_1, y_2 (over $\mathbb{Z}_q$) with matrix of coefficients equal to

$$C = \begin{pmatrix} 1 & r & 0 & 0 \\ 0 & 0 & 1 & r \\ r'' & rr''' & \alpha r'' & \alpha rr''' \end{pmatrix}$$

which has rank 3 since $r'' \neq r'''$ (the considered query is illegal).

Cramer-Shoup: Security Proof

Each $\mu \in \mathbb{G}$ is a possible value for $u_1^{x_1+\alpha y_1} u_2^{x_2+\alpha y_2}$.

We have q^2 possible values for x_1, x_2, y_1, y_2 from (4), (5).

The map sending a possible value (x_1, x_2, y_1, y_2) in $u_1^{x_1+\alpha y_1} u_2^{x_2+\alpha y_2}$ is surjective (with the range bein $\mathbb{G}$), and the preimage of each $\mu \in \mathbb{G}$ contains q distinct elements.

Fixed u_1, u_2, e, the adversary $\mathcal{A}$ cannot predict the value of $u_1^{x_1+\alpha y_1} u_2^{x_2+\alpha y_2}$ with probability better than $1/q$.

Cramer-Shoup: Security Proof

If the first illegal decryption query (u_1, u_2, e, v) is rejected, $\mathcal{A}$ learns that $v \neq u_1^{x_1+\alpha y_1} u_2^{x_2+\alpha y_2}$.

This eliminates 1 of q possibile values for v.

The probability that the $\ell(n)$-th decryption query of this form is not rejected is at most $1/(q - (\ell(n) - 1))$.

Thus the probability that one of these queries is not rejected is at most $\ell(n)/(q - (\ell(n) - 1))$, which is negligible in n (q is exponential in n, $\ell(n)$ is polynomial).

Cramer-Shoup: Security Proof

After the challenge ciphertext is sent

From the challenge ciphertext $CT^* = (u_1^*, u_2^*, e^*, v^*)$, $\mathcal{A}$ learns:

$$\log_{g_1} v^* = (x_1 + \alpha^* y_1)k + (x_2 + \alpha^* y_2)rr'. \tag{7}$$

Cramer-Shoup: Security Proof

After the challenge ciphertext is sent

From the challenge ciphertext $CT^* = (u_1^*, u_2^*, e^*, v^*)$, $\mathcal{A}$ learns:

$$\log_{g_1} v^* = (x_1 + \alpha^* y_1)k + (x_2 + \alpha^* y_2)rr'. \tag{7}$$

We have three possible types of illegal queries (u_1, u_2, e, v):

- $(u_1, u_2, e) = (u_1^*, u_2^*, e^*)$ with $v \neq v^*$. Since the hash values are equal but $v \neq v^*$, the decryption oracle rejects.
- $(u_1, u_2, e) \neq (u_1^*, u_2^*, e^*)$ and $\alpha = \alpha^*$. It means a collision in H has been found. But H is collision-resistant, so this happens only with negligible probability.

Cramer-Shoup: Security Proof

- $(u_1, u_2, e) \neq (u_1^*, u_2^*, e^*)$ and $\alpha \neq \alpha^*$. The decryption oracle accepts the query only if

$$\log_{g_1} v = (x_1 + \alpha y_1)\tilde{r} + (x_2 + \alpha y_2) r \tilde{r}' \tag{8}$$

where $\tilde{r} = \log_{g_1} u_1 \neq \tilde{r}' = \log_{\tilde{g}_2} u_2$.

Cramer-Shoup: Security Proof

- $(u_1, u_2, e) \neq (u_1^*, u_2^*, e^*)$ and $\alpha \neq \alpha^*$. The decryption oracle accepts the query only if

$$\log_{g_1} v = (x_1 + \alpha y_1)\tilde{r} + (x_2 + \alpha y_2) r\tilde{r}' \qquad (8)$$

where $\tilde{r} = \log_{g_1} u_1 \neq \tilde{r}' = \log_{\tilde{g}_2} u_2$.

In this case, the equations (4), (5), (7) and (8) are linearly independent because

$$\det \begin{pmatrix} 1 & r & 0 & 0 \\ 0 & 0 & 1 & r \\ k & r'r & k\alpha^* & rr'\alpha^* \\ \tilde{r} & r\tilde{r}' & \tilde{r}\alpha & r\tilde{r}'\alpha \end{pmatrix} = (r^2)(r' - k)(\tilde{r} - \tilde{r}')(\alpha - \alpha^*) \neq 0.$$

Cramer-Shoup: Security Proof

We have q possible values for x_1, x_2, y_1, y_2 from (4),(5),(7). For each of them, only one value of $v \in \mathbb{G}$ makes D decrypt.

Fixed u_1, u_2, e, $\mathcal{A}$ cannot predict the value of $u_1^{x_1+\alpha y_1} u_2^{x_2+\alpha y_2}$ with probability better than $1/q$.

If the first illegal decryption query (u_1, u_2, e, v) is rejected, $\mathcal{A}$ learns that $v \neq u_1^{x_1+\alpha y_1} u_2^{x_2+\alpha y_2}$.

This eliminates 1 of q possibile values for v.

The probability that the $\ell(n)$-th decryption query of this form is not rejected is at most $1/(q-(\ell(n)-1))$.

Thus the probability that one of these queries is not rejected is at most $\ell(n)/(q-(\ell(n)-1))$, which is negligible in n (q is exponential in n, $\ell(n)$ is polynomial). $\square$

Dlog-based Collision-Resistant Hash Functions

Theorem

If the discrete logarithm is hard for some group generation algorithm $\mathcal{G}$, then collision-resistant hash functions exist.

Dlog-based Collision-Resistant Hash Functions

Theorem

If the discrete logarithm is hard for some group generation algorithm $\mathcal{G}$, then collision-resistant hash functions exist.

Suppose $\mathcal{G}$ generates prime-order groups.

We define a fixed-length hash function (KeyGen, H) as follows:

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

  It then selects a uniform $h \in \mathbb{G}$ and outputs the key $s = (\mathbb{G}, q, g, h)$.

- $H^s(x_1, x_2) \leftarrow H(s, (x_1, x_2) \in \mathbb{Z}_q \times \mathbb{Z}_q)$: on input a key s and a pair (x_1, x_2), it outputs $H^s(x_1, x_2) := g^{x_1} h^{x_2} \in \mathbb{G}$.

Dlog-based Collision-Resistant Hash Functions

If a collision for H^s is found, the Dlog problem can be solved.

Suppose that $H^s(x_1, x_2) = H^s(x_1', x_2')$ for $(x_1, x_2) \neq (x_1', x_2')$.

Then $g^{x_1}h^{x_2} = g^{x_1'}h^{x_2'}$ and hence:

$$g^{x_1 - x_1'} = h^{x_2' - x_2} \Longrightarrow \log_g h = [(x - x_1') \cdot (x_2' - x_2)^{-1} \pmod q].$$

Note that $x_2' - x_2 \neq 0 \pmod q$, otherwise we have $x_1 = x_1'$ mod q and therefore no collision is found.

As q is prime, the inverse of $(x_2' - x_2)$ exists.

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.