*Hochschule Flensburg* (2026), 1–5

**A SUMMARY**

# The Cramer–Shoup Public-Key Encryption Scheme (CS1)

Alina Abraham, 750538

**Abstract**

Public-key encryption (PKE) allows secure communication over open networks without the need of a shared secret in advance. However, classical security notions for PKE often assume passive adversaries that only observe ciphertexts. This does not accurately reflect realistic network environments where attackers may interact with protocols and obtain partial decryption capabilities. The strongest standard security notion for PKE is indistinguishability under adaptive chosen ciphertext attacks (IND-CCA2), where an adversary can query a decryption oracle before and after receiving a challenge ciphertext. This paper provides a detailed and accessible description of the Cramer–Shoup encryption scheme (CS1), the first practical PKE construction proven IND-CCA2 secure under standard assumptions without relying on the random oracle model. This paper includes the required background, the relevant security notions, a formal description of CS1 and the main soundness and security arguments based on the Decisional Diffie–Hellman (DDH) assumption and target collision resistant hashing (TCR).

**Keywords:** Public-Key Encryption, IND-CCA2, Cramer–Shoup, Decisional Diffie–Hellman, Target Collision Resistance

## 1. Introduction

Public-key encryption (PKE) is a fundamental concept in cryptography where a pair of mathematically related keys, a public and a private key, ensure a secure communication over a open network. This makes it possible to exchange hidden messages without requiring a shared secret in advance. The classical security notions for PKE assume a passive adversary, being a hypothetical algorithms that models the attackers strategy, that can observe ciphertext but cannot interact. While this setup is sufficient in a theoretical setting, it is hard to transfer it to realistic network environments where an attacker may try obtaining decryption capabilities through protocol interactions.

In stronger models the adversary can adaptively submit ciphertexts to a decryption oracle and use the response to gain information about other encrypted messages. This setup represents a real life network environment more accurately.

There are a few encryption schemes that are secure under a passive adversary, but are completly insecure against the stronger attack model. A good example is the ElGamal encryption scheme. (*ElGamal 1985*) To prevent such attacks from breaking the scheme, the system has to remain secure even in the presence of an active adversary with decryption oracle access.

Prior to the work of Cramer and Shoup, encryption schemes only achieved security against active adversaries by relying on heuristic assumptions, such as the random oracle model. These kind of assumptions make a theoretical solution possible but cannot be implemented in the real world. The Cramer-Shoup (CS) construction was the first practical public-key encryption scheme proven secure against an active adversary or precisely against adaptive chosen ciphertext attacks (IND-CCA2). (*Cramer and Shoup 2001*)

This paper provides a detailed and accessible description of the Cramer–Shoup encryption scheme and the theoretical foundations on which its security is based.

## 2. Preliminaries

### 2.1 Notation and Mathematical Background

Throughout this paper a few cryptographically specific notations are used, that are helpful to clarify upfront.

The security parameter $\lambda \in \mathbb{N}$ is used as an input for an algorithm in the form of $1^\lambda$. The parameter makes it possible to tell the algorithm how much security is wanted. It is used to generate the keys that the scheme relies on.

Many algorithms used in this paper are probabilistic, meaning that they make use of internal randomness during their execution. This randomness is necessary for achieving security properties such as indistinguishability and semantic security (more on that later).

For random sampling we use the notation $x \leftarrow_R X$, which means that the value $x$ is chosen uniformly at random from the finite set $X$.

All cryptographic algorithms considered in this paper are probabilistic polynomial-time (PPT) algorithms, ensuring that they can be efficiently executed while still providing the necessary security guarantees.

Furthermore we work in cyclic groups $G$ of prime order $q$, generated by an element $g$. Group operations are written multiplicatively.

Lastly, a function $\text{negl}(\lambda)$ is called negligible if it grows slower than the inverse of any polynomial in $\lambda$.

### 2.2 Decisional Diffie–Hellman (DDH)

Cryptographic security proofs rely on assumptions about the computational difficulty of certain mathematical problems. A cryptographic assumption states that no efficient algorithm exists that can solve a given problem with more than negligible probability. Assumptions are not proven statements, but problems that haven't been solved yet. (*Katz and Lindell 2014*)

An assumption is said to be intractable if no PPT adversary can solve the corresponding problem except with negligible probability. Security proofs therefore show that breaking a

cryptographic scheme would imply the existence of a PPT algorithm that violates the underlying intractability assumption.

Assumptions can be compared in terms of strength. Informally, a stronger assumption rules out more possible attacks, while a weaker assumption rules out fewer. If assumption $A$ implies assumption $B$, but not the other way around, then $A$ is considered stronger than $B$. In cryptographic design we rely on the weakest possible assumption that proves security.

In the context of cryptographic designs based on Diffie-Hellman-based, there are three closely related assumptions: the Discrete Logarithm Problem (DLP), the Computational Diffie–Hellman problem (CDH), and the Decisional Diffie–Hellman problem (DDH). While DDH is the weakest of these assumption and therefore relies on CDH and DLP.

$$DLP \implies CDH \implies DDH$$

The discrete logarithm assumption states that, given a cyclic group $G$ of prime order $q$, a generator $g$, and an element $g^x$ for random $x \in \mathbb{Z}_q$, it is infeasible for any PPT adversary to compute the exponent $x$. This is an extraction problem, because the adversary must explicitly recover secret information.

The computational Diffie–Hellman assumption states that, given $g^x$ and $g^y$ for random $x, y \in \mathbb{Z}_q$, it is infeasible to compute the shared secret $g^{xy}$.

The decisional Diffie–Hellman (DDH) assumption is a distinguishing assumption, rather than an extraction assumption like the previous ones. Informally, DDH states that Diffie–Hellman tuples are computationally indistinguishable from random tuples. Formally, given a tuple $(g^x, g^y, g^z)$, where $x, y, z \in \mathbb{Z}_q$ are chosen uniformly at random, no PPT adversary can distinguish with non-negligible advantage whether $z = xy \bmod q$ or whether $g^z$ is an independent random group element.

All three assumptions are average-case assumptions, meaning that they assert hardness over randomly generated instances of the problem, rather than worst-cases. This is crucial for cryptographic applications, since keys and ciphertexts are generated randomly and an assumption may fail on one special, specific instances while still holding on average. As a result, such failures do not invalidate the security guarantees. (*Katz and Lindell 2014*)

The Cramer–Shoup encryption scheme relies specifically on the DDH assumption. This is because its security goal is indistinguishability under chosen ciphertext attacks, and therefore it is a distinguishing problem. (*Cramer and Shoup 2001*)

### 2.3 Target Collision Resistant Hashing (TCR)

In addition to DDH, Cramer-Shoup scheme also relies on another assumption, the target collision resistence (TCR). TCR states that after a random hash key (to determine the hashing algorithm) and a random target input are fixed, it is computationally infeasible for an adversary to find a different input that produces the same hash value. (*Cramer and Shoup 2001; Katz and Lindell 2014*)

TCR also relies on the stonger standard collision resistance, where the adversary may choose both colliding inputs freely.

$$CR \implies TCR$$

The Cramer–Shoup scheme, uses TCR hashing to bind together the components of a ciphertext. This prevents an adversary from modifying a valid ciphertext without detection.

## 3. Secure Public-Key Encryption

### 3.1 Public-Key Encryption (PKE)

A public-key encryption scheme consists of three algorithms: a key generation algorithm KeyGen, an encryption algorithm Encrypt, and a decryption algorithm Decrypt. Katz and Lindell 2014

- The key generation algorithm outputs a public key $PK$ and a related secret key $SK$.

$$\text{KeyGen}(1^k) \to \text{PK, SK}$$

- The encryption algorithm takes the public key $PK$ and a message $m$ as input and outputs a ciphertext $\psi$.

$$\text{Enc}(\text{PK}, m) \to c$$

- The decryption algorithm takes the secret key $SK$ and a ciphertext $\psi$ and outputs either the original message $m$ or a rejection symbol.

$$\text{Dec}(\text{SK}, c) \to m$$

The purpose of public-key encryption is to allow any party to encrypt messages using the public key, while only the holder of the secret key can decrypt them.

### 3.2 Soundness

Soundness, sometimes referred to as correctness, ensures that the encryption and decryption algorithms are compatible. Intuitively, if all parties follow the protocol, so behave honestly, the system should work as expected and decrypting an encrypted message with the corresponding keys should result in the original message. (*Katz and Lindell 2014*)

$$\text{Dec}(\text{SK}, \text{Enc}(\text{PK}, m)) \to m$$

Requiring perfect correctness for all keys and all randomness is often too strong. Instead, the notion of soundness used by Cramer and Shoup allows for a negligible probability that the key generation algorithm produces a bad key pair. A key pair is called bad if there exists a message and a corresponding ciphertext that fails to decrypt correctly. So the scheme is sound if the probability of generating such a bad key pair is negligible. (*Cramer and Shoup 2001*)

### 3.3 Security Against Adaptive Chosen Ciphertext Attacks (CCA2)

Security notions for public-key encryption differ in the amount of power they grant to an adversary. Early notions, such as semantic security and indistinguishability under chosen-plaintext attacks (IND-CPA), assume a passive adversary who can obtain encryptions of messages of its choice but cannot interact with the decryption process. While these notions are sufficient to guarantee confidentiality in restricted settings, they do not really capture many realistic attack scenarios in which an adversary may actively manipulate ciphertexts. (*Katz and Lindell 2014*)

Semantic security informally states that a ciphertext should not reveal any partial information about the underlying plaintext. This notion is equivalent to IND-CPA security, which is defined via an indistinguishability experiment (see Figure 1).

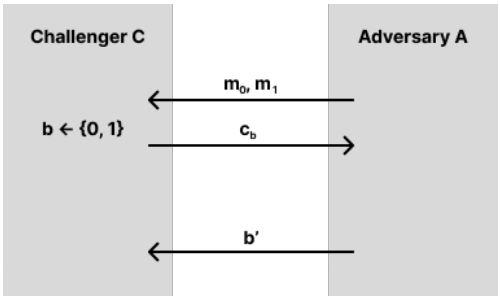


**Figure 1.** IND-CPA Experiment

The adversary chooses two messages $m_0$ and $m_1$, receives an encryption of one of them, and attempts to guess which message was encrypted. A scheme is IND-CPA secure if no probabilistic polynomial-time (PPT) adversary can succeed with more than negligible advantage.

$$\Pr[b' = b] \leq 0,5 + \text{negl}.$$

However, IND-CPA security does not prevent an adversary from modifying ciphertexts in meaningful ways. This property is called malleability and it refers to the ability of an adversary to transform a ciphertext into another ciphertext, e.g. in multiplicatively homomorphic schemes such as ElGamal where the decrypted message is multiplied by a known factor. (*ElGamal 1985*) Although this does not violate IND-CPA security, it becomes problematic in the presence of a decryption oracle, since the adversary may exploit such transformations to gain information about the original plaintext.

To model such active attacks, stronger security notions are required. In a chosen ciphertext attack (CCA), the adversary is given access to a decryption oracle in addition to the encryption functionality. Two variants of CCA security are commonly distinguished. In CCA1 security (also called non-adaptive CCA), where the adversary may interact with the decryption oracle only before receiving the challenge ciphertext (see Figure 2). This already captures some forms of active attacks, it does not allow the adversary to adapt its decryption queries based on the challenge though.

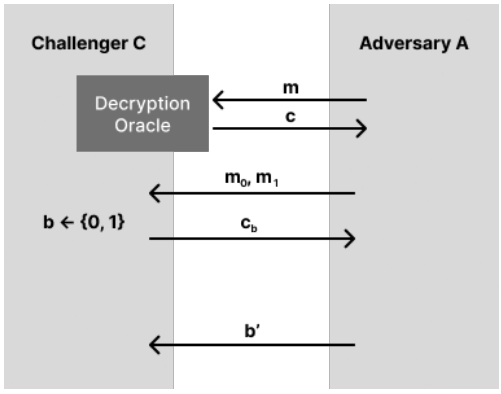


**Figure 2.** IND-CPA Experiment

The strongest and most realistic notion is adaptive chosen ciphertext security (CCA2). In the IND-CCA2 experiment, the adversary is allowed to query the decryption oracle both before and after receiving the challenge ciphertext, only with the restriction that it may not decrypt the challenge ciphertext itself (see Figure 3).

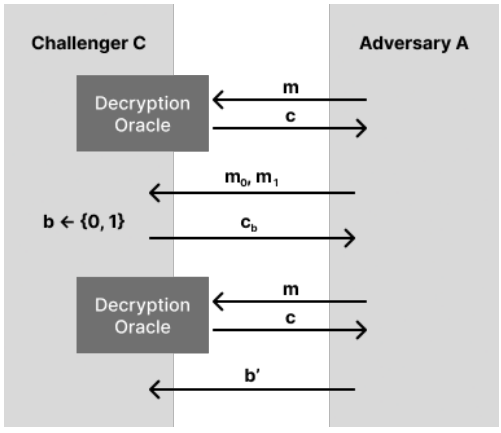


**Figure 3.** IND-CPA Experiment

This models an adversary that can interact arbitrarily with a system even after observing the target ciphertext. A public-key encryption scheme is said to be IND-CCA2 secure if any PPT adversary can distinguish encryptions of chosen messages only with negligible advantage over random guessing.

IND-CCA2 security is widely regarded as the strongest standard notion of confidentiality for public-key encryption and it implies semantic security, IND-CPA security, and non-malleability. (*Katz and Lindell 2014* ) Nevertheless, achieving IND-CCA2 security is challenging, as it requires to ensure message indistinguishability and prevent any meaningful manipulation of ciphertexts.

## 4. The Scheme CS1

### 4.1 Description of the Scheme

The Cramer–Shoup encryption scheme is based on the ElGamal encryption scheme but introduces additional structure to prevent chosen ciphertext attacks. (*ElGamal 1985; Cramer and*

*Shoup 2001*) The public key consists of multiple group elements derived from secret exponents, while the secret key contains these exponents. Encryption uses randomness to generate Diffie–Hellman values and masks the message multiplicatively, similar to ElGamal.

In addition, the ciphertext includes a hash-based verification component that binds together all ciphertext elements. During decryption, this verification is checked before the message is recovered. If the verification fails, the ciphertext is rejected. This mechanism ensures that modified ciphertexts are detected and prevents adversaries from exploiting the decryption oracle.

A formal description of CS1 looks like this:

*Parameters and Setup*

- Let $\lambda$ be the security parameter.
- Let $\Gamma = (\hat{G}, G, g, q)$ be a group description:
  - $G \subseteq \hat{G}$ cyclic of prime order $q$
  - $g$ generator of $G$
- Let `HF` be a target collision resistant (TCR) hash family.

*Key Generation*

Input: $1^\lambda$

1. Sample a group description:
   $\Gamma = (\hat{G}, G, g, q) \leftarrow_R \hat{S}(1^\lambda)$
2. Sample a hash key:
   $hk \leftarrow_R HF.\text{KeySpace}_{\lambda,\Gamma}$
3. Sample secret exponents:
   $w \leftarrow_R \mathbb{Z}_q^*, \quad x_1, x_2, y_1, y_2, z_1, z_2 \leftarrow_R \mathbb{Z}_q$
4. Compute public group elements:
   $\hat{g} = g^w,\ e = g^{x_1}\hat{g}^{x_2},\ f = g^{y_1}\hat{g}^{y_2},\ h = g^{z_1}\hat{g}^{z_2}$

Output:

- Public key:
  $PK = (\Gamma, hk, \hat{g}, e, f, h)$
- Secret key:
  $SK = (\Gamma, hk, x_1, x_2, y_1, y_2, z_1, z_2)$

*Encryption*

Input: public key $PK$, message $m \in G$

1. Sample randomness:
   $u \leftarrow_R \mathbb{Z}_q$
2. Compute Diffie–Hellman components:
   $a = g^u, \quad \hat{a} = \hat{g}^u$
3. Mask the message:
   $b = h^u, \quad c = b \cdot m$
4. Compute hash value:
   $v = HF_{hk}^{\lambda,\Gamma}(a, \hat{a}, c)$
5. Compute verification component:
   $d = e^u \cdot f^{uv}$

Output ciphertext:
$\psi = (a, \hat{a}, c, d)$

*Decryption*

Input: secret key $SK$, ciphertext $\psi = (a, \hat{a}, c, d)$

1. Syntax check:
   - reject if $\psi$ is not a 4-tuple
2. Group membership check:
   - reject if $a, \hat{a}, c \notin G$
3. Recompute hash $v = HF_{hk}^{\lambda,\Gamma}(a, \hat{a}, c)$
4. Verification check:
   - accept only if $d = a^{x_1+y_1v} \cdot \hat{a}^{x_2+y_2v}$
5. Recover masking term $b = a^{z_1} \cdot \hat{a}^{z_2}$
6. Recover message $m = c \cdot b^{-1}$

Output: message $m$ or `reject`

### 4.2 Soundness Analysis

Soundness of the Cramer–Shoup scheme follows from the structure of the encryption and decryption algorithms. For honestly generated keys and ciphertexts, the verification equation checked during decryption always holds, and the masked message can be correctly recovered. The probability that key generation produces a key pair for which this property does not hold is negligible. Therefore, the scheme is sound according to the definition given in the preliminaries.

*Theorem*

Let $(PK, SK) \leftarrow \text{KeyGen}(1^\lambda)$ and let $m \in G$ be any message.
Let $\psi \leftarrow \text{Encrypt}(PK, m)$.
Then $\Pr[\text{Decrypt}(SK, \psi) = m] = 1$ except with negligible probability over the randomness of KeyGen.

*Proof*

Let the public and secret keys be generated as specified by the scheme:

- $\hat{g} = g^w$
- $e = g^{x_1}\hat{g}^{x_2}$
- $f = g^{y_1}\hat{g}^{y_2}$
- $h = g^{z_1}\hat{g}^{z_2}$

During encryption, a random value $u \leftarrow_R \mathbb{Z}_q$ is chosen and the ciphertext $\psi = (a, \hat{a}, c, d)$ is computed as:

$$a = g^u, \quad \hat{a} = \hat{g}^u, \quad b = h^u, \quad c = b \cdot m$$

$$v = HF_{hk}^{\lambda,\Gamma}(a, \hat{a}, c), \quad d = e^u \cdot f^{uv}$$

**Verification Step**

During decryption, the algorithm recomputes

$$v = HF_{hk}^{\lambda,\Gamma}(a, \hat{a}, c)$$

and checks whether $d \stackrel{?}{=} a^{x_1+y_1v} \cdot \hat{a}^{x_2+y_2v}$

Substituting the definitions of $a$ and $\hat{a}$, we get:

$$a^{x_1+y_1v} \cdot \hat{a}^{x_2+y_2v} = (g^u)^{x_1+y_1v} \cdot (\hat{g}^u)^{x_2+y_2v}$$

$$= g^{ux_1} \cdot \hat{g}^{ux_2} \cdot g^{uy_1v} \cdot \hat{g}^{uy_2v}$$
$$= \left(g^{x_1}\hat{g}^{x_2}\right)^u \cdot \left(g^{y_1}\hat{g}^{y_2}\right)^{uv}$$
$$= c^u \cdot f^{uv}$$

which is exactly the value of $d$ computed during encryption. Therefore, the verification check in decryption always succeeds for honestly generated ciphertexts.

**Message Recovery**

After successful verification, the decryption algorithm computes:

$$b = a^{z_1} \cdot \hat{a}^{z_2}$$

Substituting $a = g^u$ and $\hat{a} = \hat{g}^u$, we get:

$$b = \left(g^u\right)^{z_1} \cdot \left(\hat{g}^u\right)^{z_2} = \left(g^{z_1}\hat{g}^{z_2}\right)^u = h^u$$

Finally, the message is recovered as:

$$m = c \cdot b^{-1}$$

Since $c = b \cdot m$, this yields:

$$m = (b \cdot m) \cdot b^{-1} = m$$

For any honestly generated key pair and any message $m \in G$, decryption of an honestly generated ciphertext always result in the original message. Therefore, the Cramer–Shoup encryption scheme is sound, except with negligible probability over key generation. □

### *4.3 Security Analysis*

To prove the security we can rely on our security assumptions. We approach this by proving that if the scheme were false then out assumptions had to be false as well. This is done iteratively through a sequence of games, each differing from the previous one by a small change.

*Theorem*

CS1 is IND-CCA2 secure assuming the hardness of DDH in G and TCR of HF.

*Proof*

Let $A$ be any PPT adversary against CS1 in the IND-CCA2 experiment.

**Game 0**

This is the real IND-CCA2 experiment against CS1.

**Game 1**

Is a hypothetical game, that is exactly like Game 0 only that we modify the description oracle so it rejects **any** ciphertext that is different from the challenge ciphertext. This modification exposes an Event $F$, which can be defined as followed: The adversary $A$ submits a ciphertext $(a, \hat{a}, c, d) \neq (a*, \hat{a}*, c*, d*)$ that passes the verification checks in decryption. Our modification artificially blocks this event $F$.

As a logical result of this we can now say, that if Game 0 and Game 1 were noticeably different then this would mean that Event $F$ happens often implying that it is possible for the adversary to find another ciphertext that passes verification but is different from the challenge. This would break TCR.

Finally we can say that assuming TCR holds, the difference between Game 0 and Game 1 is negligible.

$$|\Pr[G0] - \Pr[G1]| \rightarrow \text{negligible (TCR)}$$

**Game 2**

Is a hypothetical game, that is exactly like Game 1 only that we replace the masking of the ciphertext $h^u$ with a uniform random group element $R$. This means when in Game 1 $c$ was computed as $c = m * h^u$ in Game 2 $c$ is now $c = m * R$ with $R \leftarrow_R G$.

We can draw a similar conclusion here: if Game 1 were noticeably different from Game 2 then the adversary could distinguish $h^u$ from $R$ and therefore breaking DDH.

Assuming DDH holds, the difference between Game 1 and Game 2 is negligible.

$$|\Pr[G1] - \Pr[G2]| \rightarrow \text{negligible (DDH)}$$

**Game 3**

This is the trivial game that is used to put everything together and finalize the proof. We can say that in this game the ciphertext is generated independently of the challenge bit, which is equivalent to Game 2.

$$|\Pr[G2] - \Pr[G3]| = 0$$

Since the ciphertext is independent of the challenge bit, the adversaries advantage is negligible and hence the probability of winning the game is $\frac{1}{2}$ + negl. This proves CS1's security. □

## 5. Conclusion

This paper was a summary and approachable explanation of the Cramer–Shoup public-key encryption scheme CS1 together with the theoretical background required to understand its security guarantees. Overall, the CS1 scheme demonstrates how CCA2 security can be achieved under standard assumptions without relying on the random oracle model. This makes the Cramer–Shoup construction a foundational result in modern public-key cryptography.

## References

Cramer, Ronald, and Victor Shoup. 2001. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *IACR Cryptology ePrint Archive,* 108. http://eprint.iacr.org/2001/108.

ElGamal, Taher. 1985. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory* 31 (4): 469–472. https://doi.org/10.1109/TIT.1985.1057074.

Katz, Jonathan, and Yehuda Lindell. 2014. *Introduction to modern cryptography, second edition.* CRC Press. ISBN: 9781466570269. https://www.crcpress.com/Introduction-to-Modern-Cryptography-Second-Edition/Katz-Lindell/p/book/9781466570269.