Balanced Security and Privacy Protection in Digital Content Distribution Systems

Antonius Cahya Prihandoko[1,*], Stanislaus Jiwandana Pinasthika[2], Hossein Ghodosi[3]
[1,2]Informatics Dept, Faculty of Computer Science, Universitas Jember, Indonesia
[3]Information Technology Dept, James Cook University, Townsville QLD, Australia

Email: [1]antoniuscp.ilkom@unej.ac.id, [2]stanislausjp@unej.ac.id, [3]hossein.ghodosi@jcu.edu.au

DOI: https://doi.org/10.31284/j.jtm.2025.v6i2.7800
Received June 7th 2025; Received in revised June 20th 2025; Accepted June 23rd 2025; Available online July 9th 2025

Abstract

Security protection for content providers is essential in a digital content distribution system so that only authorized users can access the content. However, focusing on the security aspect often makes the system ignore the privacy of content users. This article presents a model of protocol that can provide balanced protection of content provider security and user privacy in a digital content distribution system. This protocol is based on oblivious transfer (OT), a standard protocol in cryptography that allows the sender of a message to send a certain amount of information securely to the recipient of the message, such that at the end of the protocol the recipient of the message cannot access more information than specified, while the sender of the message cannot know which information was successfully accessed by the recipient. Assuming the existence of tamper-proof devices, the protocol presented in this article can provide excellent protection for both the security of content providers and the privacy of content users.

Keywords: *digital content distribution system; oblivious transfer; content provider security; user privacy.*

1. Introduction

Secure content delivery from providers to users is an essential aspect of a digital content distribution system. Ensuring secure distribution guarantees that only authorized users can access the content. In cryptographic approaches, content is encrypted before distribution to maintain security. To enhance protection, some cryptographic research focuses on encryption-decryption key management [1][2][3][4], while others modify encryption algorithm implementations to make them harder for hackers to decipher, such as code obfuscation [5][6] and white-box cryptography [7][8]. These methods ultimately aim to keep decryption keys confidential, ensuring that users must obtain proper licenses to decrypt and use protected content correctly.

However, an excessive focus on content security often leads to neglecting user privacy. Systems typically collect personal user data to allocate appropriate content usage rights, but users lack transparency regarding how and when content providers use their data. As a result, user privacy is frequently overlooked or even sacrificed. From the content provider's perspective, acquiring user data is crucial for understanding purchasing patterns and accelerating sales and profit targets. Providers may use this data for marketing purposes without user consent, further violating privacy and reducing user satisfaction. Therefore, protecting user privacy must also be taken seriously.

Privacy protection approaches aim to minimize user data acquisition [9]. In practice, systems avoid linking user identities to accessed items, similar to anonymous cash [10][11] and blind decryption methods [12][13]. In the anonymous cash approach, the provider knows which items were purchased but not who purchased them. However, this method risks fraudulent activities, such as users spending a token multiple times. In blind decryption, the provider knows who made the purchase but not what was bought, though item identities may still be inferred if different items have distinct prices.

[*]*antoniuscp.ilkom@unej.ac.id*

These approaches provide only partial solutions—either securing content providers or protecting user privacy. The challenge is how a digital content distribution system can offer balanced protection for both. To address this issue, this paper presents a content distribution protocol model that comprehensively ensures both content provider security and user privacy. The protocol is built on oblivious transfer (OT) [14], a cryptographic protocol that enables two parties to privately exchange one or more secret messages. An OT protocol must be designed to ensure security for the sender and privacy for the recipient [15]. Security for the sender means the recipient cannot access more information than specified, while privacy for the recipient means the sender does not know which message was accessed. Given these characteristics, OT has the potential to be applied in building a secure and private content distribution system [16][17].

2. Methodology

To achieve balanced protection for both content provider security and user privacy in a digital content distribution system, our protocol is structured into three main stages: (1) protocol construction, (2) implementation, and (3) security and privacy analysis.

At its core, the protocol is built on *Oblivious Transfer (OT)* and *Shamir's Secret Sharing*. OT enables selective transmission of information such that the sender remains unaware of which piece of data the receiver accessed, while the receiver gains no access to other data besides the selected part. Shamir's scheme allows a secret (e.g., a decryption key) to be split into multiple parts, such that only a minimum number of parts are needed to reconstruct it.

Overview of the Transaction:

Let us suppose:

- A content provider owns n encrypted digital items.
- A user is licenced to access k of these n items, where $k < n$.

The protocol works as follows:

1. **Content Encryption**: The provider encrypts each item using a symmetric key S, then splits S into n shares using Shamir's Secret Sharing [18] with a threshold of $(n–k)$. The division of S into multiple parts S_i follows a polynomial of degree $(n - k - 1)$ as shown in Equation (1):

   $$f(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-k-1} x^{n-k-1} \quad (1)$$

   where $S = a_0$ and $S_i = f(i)$. All arithmetic operations are performed modulo a prime number p greater than S and n. The coefficients in $f(x)$ are randomly selected from a uniform distribution over integers in $[0, p)$.

2. **Share Embedding**: Each share is paired with its respective encrypted content and embedded into a tamper-proof smart card. These smart cards contain two linked functions per item:
   - GetKey (GK_i): Retrieves key share S_i.
   - GetContent (GC_i): Decrypts content M_i using the reconstructed key.

3. **Execution Flow**:
   - The user executes $(n - k)$ GetKey functions to collect enough shares and reconstruct S using ***Lagrange interpolation*** as shown in Equation (2):

   $$f(x) = \sum_{i=1}^{n-k} y_i \prod_{j=1, j\neq i}^{n-k} \frac{x - x_j}{x_i - x_j} \quad (2)$$

- The remaining *k* content items are then accessed using their corresponding GetContent functions.

In simpler terms: to access any item, the user must sacrifice others. The protocol ensures that retrieving the key makes the forfeited items permanently inaccessible—enforcing strict access without tracking user activity.

The protocol implementation is modeled using smart cards. A smart card with an embedded microprocessor is used to store and process data, including secret key fragments and associated content values, as well as functions for key reconstruction and content access. The security and privacy analysis is then conducted based on the mechanisms executed by the content distribution protocol.

3. Results and Discussion

3.1 Content Distribution Protocol Construction

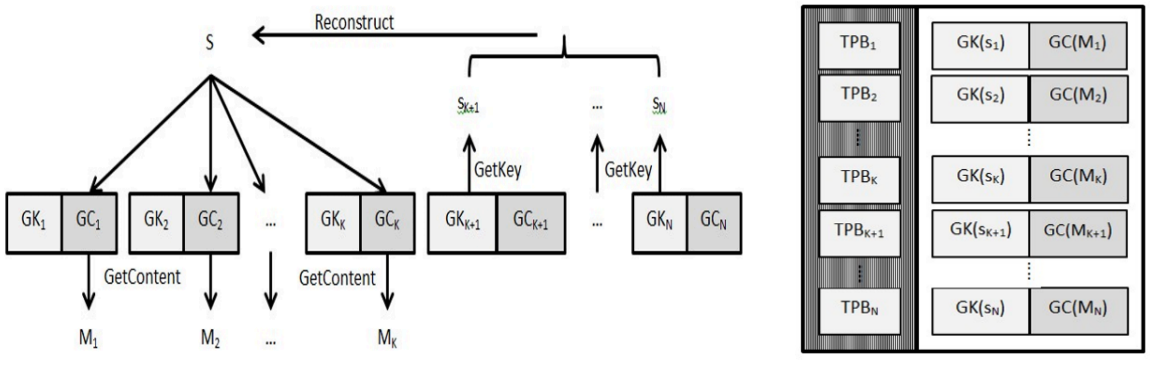
The constructed content distribution protocol utilizes tamper-proof devices, which are designed to execute only once before becoming inaccessible. These devices contain pairs of functions: GetKey (GK) and GetContent (GC). The GK function retrieves decryption key fragments, while the GC function accesses content using the reconstructed decryption key. Within each function pair, only one function can be executed—once GK is used, its corresponding GC function becomes inaccessible. This mechanism ensures that content providers can securely distribute content to users while maintaining privacy.

For instance, suppose the content provider (Alice) offers *n* content items ($M_1, M_2, ..., M$□), and the user (Bob) wishes to access *k* items, where $k < n$. Alice holds a secret key *S* for content access and applies Shamir's secret sharing scheme with a threshold of *(n - k)* to split *S* into *n* fragments. This ensures that at least *(n - k)* fragments are required to reconstruct *S*.

The protocol follows these steps:

1. Alice selects a prime number *p* greater than *n* and randomly chooses *(n - k - 1)* elements from *Z*□ to construct a polynomial $f(x) = S + a_1x^1 + a_2x^2 + ... + a_{n-k-1}x^{n-k-1}$
2. For each *i* in *{1, 2, ..., n}*, Alice computes $S_i = f(i)$.
3. Each tamper-proof device stores a key fragment S_i associated with content M_i, accessible via functions GK_i and GC_i.
4. The tamper-proof devices containing key fragments and content values are provided to Bob.

After sending the device, there was no further communication between Alice and Bob. After receiving the devices, Bob can access *k* content items by sacrificing *(n - k)* items. The protocol ensures that Bob reconstructs *S* using *(n - k)* key fragments, preventing access to the remaining *(n - k)* items.



(a) (b)

Figure 1. a) Process of obtaining k from n content items, b) Smart card model.

For simplicity, suppose the k content items that Bob wants to access are $M_1, M_2, ..., M\square$. The detailed protocol for Bob to access these k content items (illustrated in Figure 1(a)) is as follows:

1. Bob executes $(n - k)$ GK functions, namely $GK_{k+1}, GK_{k+2}, ..., GK_n$, to obtain the key fragments $s_{k+1}, s_{k+2}, ..., s_n$.
2. Bob reconstructs the polynomial $f(x)$ from the key fragments $s_{k+1}, s_{k+2}, ..., s_n$ using Lagrange interpolation (see Equation (2)).
3. Based on the reconstructed polynomial $f(x)$, Bob can determine the decryption key S. Using the key S, Bob executes the functions $GC_1, GC_2, ..., GC_k$ to access the content items $M_1, M_2, ..., M_k$.

Figure 1(a) illustrates the process by which a user accesses k content items from a pool of n. Each tamper-proof module (TPBi) holds a key fragment and its corresponding content. The user activates $n–k$ GK functions to collect shares needed to reconstruct the decryption key S. These fragments, once used, disable their associated GC functions. After key recovery, the user activates GC functions only on the desired k items. This mechanism ensures that content is both unlocked securely and privacy-respectfully.

3.2 Smart Card Model for Protocol Implementation

The implementation of the content distribution protocol based on oblivious transfer utilizes a smart card model. A smart card is embedded with a microprocessor, allowing it to function not only as a data storage device but also as a processing unit [19].

Suppose the content provider has n content items ($M_1, M_2, ..., M\square$). First, the provider encrypts all content using a secret key S. For a given value k, where $1 \leq k \leq n - 1$, the provider splits S into n fragments ($s_1, s_2, ..., s_n$) using Shamir's secret sharing scheme with a threshold of $(n - k)$. The provider then delivers the protected content to the distributor and key fragments to the smart card manufacturer.

The protocol implementation leverages smart cards embedded with microprocessors, enabling secure storage and processing of data. The smart card model (illustrated in Figure 1(b)) follows these principles:

1. Each smart card contains n function pairs $(GK(s_i), GC(M_i))$, where $i = 1, 2, ..., n$.
2. Only one function in each pair can be executed, enforced through a one-time program (OTP) mechanism using Tamper Proof Bits (TPB).
3. To access k content items, the user executes $(n - k)$ GK functions to retrieve key fragments, reconstructs S, and then uses S to unlock k content items via GC functions.

Practically, users download protected content from a distributor and purchase a corresponding smart card. To access the content, the user connects their device to a compatible smart card reader. **Figure 1(b)** offers a simplified schematic of how smart card architecture maps the GK and GC function pairs across items, highlighting the tamper-proof one-time execution enforced by Tamper Proof Bits (TPBs). Only one function in each pair can be used, guaranteeing either access to the key fragment or to the encrypted content—but not both. Consider expanding the legend or caption to briefly explain TPB, the choice of GK vs GC, and the irreversible sacrifice required in the unlocking flow. This makes the mechanics easier to grasp without diving into equations.

3.3 Security and Privacy Analysis

Our protocol ensures robust theoretical protection. However, its security and privacy guarantees must also be examined in practical contexts.

Content Provider Security:

- **Tamper Resistance**: Assuming smart cards cannot be reverse engineered, the provider's content is shielded. Since the key *S* is never directly shared but reconstructed from disposable fragments, unauthorized users cannot access more than intended.
- **One-time Access Control**: By executing GetKey for (n–k) unused items, the corresponding GetContent functions for those items are permanently disabled. This enforces access revocation at the hardware level.

User Privacy:

- **Anonymity**: No further communication exists between the user and the provider after smart card issuance. The provider cannot trace which content was accessed.
- **Data Minimization**: No personal user data is tied to content access—purchases are done anonymously through generic smart cards.

Potential Threats and Mitigations:

The following table summarizes potential threat scenarios that may compromise the integrity or confidentiality of the proposed content distribution protocol and outlines corresponding mitigation strategies to strengthen its resilience against practical attacks.

Table 1. Potential Threats and Mitigations

Threat Scenario	Potential Weakness	Possible Defense
Smart card cloning or tampering	Unauthorized duplication or key extraction	Embed Physical Unclonable Functions (PUFs), tamper-evident seals, runtime integrity checks
Exhaustive key recovery attack	Accumulation of < (n–k) shares via multiple purchases	Randomized share generation per session/card, enforce strict uniqueness and access count
Replay or side-channel attacks	Extraction via electromagnetic or power analysis	Shielded microcontroller designs, power noise injection, secure memory execution

These safeguards, if properly implemented, can uphold the proposed privacy-preserving enforcement even under more aggressive real-world conditions.

4. Open Problem and Future Work

4.1 Limiting Content Use and Preventing Redistribution

While the current protocol guarantees that users access only the content they have paid for, it does not regulate how often or for how long that content can be used after decryption. Furthermore, it assumes good-faith usage, without enforcing controls against duplication or redistribution. To address this gap, the following extensions are proposed:

a) **Playback Limits**: Time-based or count-based access could be implemented via smart card counters or cryptographic time-lock puzzles. For example, the GC function could self-destruct or deny access after a predefined number of plays.

b) **Anti-Redistribution Protections**:

- *Watermarking/Fingerprinting*: Embed imperceptible identifiers during playback, linking content copies to the original card instance or purchase event.
- *Trusted Execution Environments (TEEs)*: Content is decrypted and played only within hardware-protected zones, preventing user-side copying.

Each approach introduces trade-offs:

- Time-based schemes require real-time clocks or network synchrony.
- Fingerprinting poses privacy questions if not properly anonymized.
- TEEs increase hardware dependency and deployment costs.

4.2 Scalability and Integration

For large-scale applications (e.g., subscription-based digital libraries or academic publishers), the current approach can be extended in two key directions:

a) **Smart Card Pooling**: Instead of issuing a unique card per transaction, a single multi-session card could support dynamic content rights encoding, possibly with reprogrammable secure elements.
b) **Protocol Abstraction**: The OT and secret-sharing layers can be encapsulated as a service layer or module, integrated into existing DRM platforms. This would ease the adoption barrier and reduce overhead for providers.

5. Conclusion

This paper proposed a cryptographic protocol that addresses the often conflicting goals of protecting content provider security and ensuring user privacy in digital content distribution systems. By integrating *Oblivious Transfer* and *Shamir's Secret Sharing* within a tamper-proof smart card framework, the protocol enforces selective content access while preserving post-purchase anonymity. The protocol guarantees that users can only access the content they are entitled to—nothing more—by sacrificing access to non-selected items. Simultaneously, the use of tamper-proof devices and non-interactive transactions ensures the provider has no visibility into user behavior, satisfying key privacy principles. Future development should address mechanisms to:

- Regulate frequency and duration of content usage (e.g., time-based access control).
- Prevent unauthorized duplication and redistribution (e.g., fingerprinting or TEE-based playback).
- Enhance scalability for large-scale platforms with dynamic licensing models.

By building a technically grounded, privacy-respecting foundation for digital content delivery, this model may provide a viable alternative to current DRM systems—especially in educational, entertainment, and licensing-intensive sectors.

References

[1] S. R. Moosavi, E. Nigussie, S. Virtanen, and J. Isoaho, "Cryptographic key generation using ECG signal," in *14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, NV, 2017, pp. 1024–1031. doi: 10.1109/CCNC.2017.7983280.

[2] R. Mahendran and K. Mani, "Generation of Key Matrix for Hill Cipher Encryption Using Classical Cipher," in *World Congress on Computing and Communication Technologies (WCCCT*, Tiruchirappalli, 2017, pp. 51–54. doi: 10.1109/WCCCT.2016.22.

[3] Y. Song, H. Wang, X. Wei, and L. Wu, "Efficient Attribute-Based Encryption with Privacy-Preserving Key Generation and Its Application in Industrial Cloud," *Secur. Commun. Networks*, vol. 2019, pp. 1–9, 2019, doi: 10.1155/2019/3249726.

[4] A. C. Prihandoko, Dafik, and I. H. Agustin, "Stream-keys generation based on graph labeling for strengthening Vigenere encryption," *Int. J. Electr. Comput. Eng.*, vol. 12, no. 4, pp. 3960–3969, 2022, doi: 10.11591/ijece.v12i4.pp3960-3969.

[5] B. Barak *et al.*, "On the (Im)possibility of Obfuscating Programs," *Adv. Cryptol.*, vol. 2139, no. Im, pp. 1–18, 2001, doi: 10.1007/3-540-44647-8.

[6] A. C. Prihandoko, H. Ghodosi, and B. Litow, "Obfuscation and WBC: Endeavour for Securing Encryption in the DRM Context," *Proc. 2013 Int. Conf. Comput. Sci. Inf. Technol.*, vol. CSIT-2013, pp. 150–155, 2013.

[7] S. Chow, P. Eisen, H. Johnson, and P. C. Van Oorschot, "A White-Box DES Implementation for DRM Applications," *Proc. ACM Work. Digit. Rights Manag. (DRM 2002), Lect. Notes Comput. Sci. 2696*, pp. 1–15, 2003, doi: 10.1007/b11725.

[8] A. C. Prihandoko and H. Ghodosi, "White-box implementation to advantage DRM," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 7, no. 2, 2017.

[9] F. Roesner and T. Kohno, "User-Centered Approaches to DRM and Privacy Trade-offs," *IEEE Secur. Priv.*, vol. 18, no. 2, pp. 45–51, 2020.
[10] D. Chaum, A. Fiat, and M. Naor, "Untraceable Electronic Cash," *Adv. Cryptol.*, vol. LNCS 403, pp. 319–327, 1990.
[11] M. Green and I. Miers, "Decentralized Content Access using Anonymous Credentials," in *Proceedings of the Privacy Enhancing Technologies Symposium (PETS)*, 2020.
[12] M. Al-Fayoumi and S. Aboud, "Blind Decryption and Privacy Protection," *Am. J. Appl. Sci.*, vol. 2, no. 4, pp. 873–876, 2005.
[13] A. C. Prihandoko and H. Ghodosi, "Blind Decryption for Preserving Privacy in the DRM System," *2021 Int. Conf. Comput. Sci. Inf. Technol. Electr. Eng. ICOMITEE 2021*, pp. 213–217, 2021, doi: 10.1109/ICOMITEE53461.2021.9650123.
[14] M. O. Rabin, "How To Exchange Secrets with Oblivious Transfer.," *Tech. Rep. TR-81, Aiken Comput. Lab, Harvard Univ.*, pp. 1–5, 1981, [Online]. Available: http://dm.ing.unibs.it/giuzzi/corsi/Support/papers-cryptography/187.pdf
[15] H. Ghodosi, "A General Model for Oblivious Transfer." the Sixth International Workshop for Applied PKC, Perth, Australia, pp. 79–87, 2007.
[16] H. Shulman, "Practical Considerations of Oblivious Transfer in Digital Distribution," *Cryptology ePrint Archive*. 2021.
[17] C. Paquin and G. Zaverucha, "Oblivious Transfer Extensions in Real-World Systems: Benchmarks and Optimization," in *ACM CCS Workshop on Applied Cryptography*, 2022.
[18] A. Shamir and A. Shamir, "How To Share a Secret," *Commun. ACM*, vol. 22, no. 1, pp. 612–613, 1979, doi: http://doi.acm.org/10.1145/359168.359176.
[19] Z. Chen, "Java Card Technology for Smart Cards: Architecture and Programmer's Guide," 2000.

How to cite this article:
Prihandoko A C, Pinasthika S J, Ghodosi H. Balanced Security and Privacy Protection in Digital Content Distribution Systems. *Jurnal Teknologi dan Manajemen.* 2025 Juli; 6(2):61-67. DOI: 10.31284/j.jtm.2025.v6i2.7800