# A Survey of Modern Cryptography Techniques for Digital Content Protection

Manisha[1]
[1]Department of Computer Science &Engineering,
Ganga Institute of Technology and Management, Kablana,
Jhajjar, Haryana, INDIA

***Abstract:*** **This paper presents is to introduce the concept of Cryptography, History of Cryptography , Modern Cryptography and analyze how digital content for mobile phones can be protected in an effective way in the context of OMA-DRM. The Open Mobile Alliance (OMA) provides specifications for content distribution for mobile phones. White-box cryptography (WBC), which focuses on software implementations of cryptographic primitives (such as encryption schemes). Traditionally, cryptographic primitives are designed to protect data and keys against black-box attacks. In such a context, an adversary has knowledge of the algorithm and may examine various inputs to and outputs from the system, but has no access to the internal details of the execution of a key instantiated primitive. In contrast, the goal of white-box implementations is to provide a degree of robustness against attacks from the execution environment. In such an environment, an adversary has unrestricted access to the software implementation.**

**We will analyze the problem in the white-box attack context where the attacker has total visibility into software implementation and execution. From these techniques we choose to focus on a relatively new technique: white-box cryptography. We can apply this technique to AES keys, by hiding the AES key in lookup tables. This prevents an attacker from finding secret keys in the implementation. The result is a functionally equivalent program in which the key is no longer visible. However, white-box cryptography increases the amount of storage space for the white-box tables, and it causes a performance slowdown. An application of white-box cryptography in which we split the set of white-box tables into a dynamic part and a static part. Each client has a unique set of static tables which can only be used in combination with a unique set of dynamic tables which are transmitted to him. The result is that whenever a key needs to be updated, no longer the whole set of tables needs to be updated.**

## I. INTRODUCTION

In traditional cryptography, a black-box attack describes the situation where the attacker tries to obtain the cryptographic key by knowing the algorithm and monitoring the inputs and outputs, but without the execution being visible. White-box cryptography addresses the much more severe threat model where the attacker can observe everything, can access all aspects of the target system/application, and may have the black-box knowledge of the crypto algorithm.

*Black box Attack*

- Attacker knows algorithm
- Watches inputs and outputs
- Controls input text
- No visibility of execution

*White-box Attack*

- Attacker can observe everything
- Attacker knows algorithm
- Watches inputs, outputs, intermediate calculations
- Controls input text
- Full visibility into Memory (debuggers and emulators)

The real strength of White-box cryptography is that it protects the whole cryptographic key at all times, rather than breaking the key up and revealing it only a piece at a time. From a security perspective, this ensures that the protected key remains hidden from hackers and is not susceptible to piecing back together in the clear during the attack process.

## II. CRYPTOGRAPHY

Encryption algorithms are very important for cryptography because cryptography use encryption algorithms to provide security and privacy. There are three classes of encryption algorithms. These are symmetric, asymmetric and digest algoritms.Encrytion is to provide the receiver to decrypt the information using his/her private key. The process of encrypting a message (plaintext) is called encryption and the process of decrypting ciphertext back into the plaintext is called decryption. There are three main materials for encryption: plaintext, key and encryption algorithms. The main goal in encryption is to make the relation between input (plaintext) and output (ciphertext).Symmetric algorithms also called conventional algorithm., are the algorithms using the same key pair to encrypt the data and the other (private key) to decrypt the data. A message digest is a hash value computed from a message. The digest has two imp. properties. Firstly it is not feasible to extract the message from the digest. Secondly, it is not feasible to construct another message that matches the digest. Most popular digest algorithms are MD5 (Message Digest Algorithm)[1] and SHA (Secure Hash Algorithm)[2].

## III. HISTORY OF CRYPTOGRAPHY

Before the modern era, cryptography was concerned solely with message confidentiality (i.e., encryption)—conversion of messages from a comprehensible form into an incomprehensible one and back again at the other end, rendering it unreadable by interceptors or eavesdroppers without secret knowledge (namely the key needed for decryption of that message). Encryption was used to (attempt to) ensure secrecy in communications, such as those of spies, military leaders, and diplomats. In recent decades, the field has expanded beyond confidentiality concerns to include techniques for message integrity checking, sender/receiver identity authentication, digital signatures, interactive proofs and secure computation, among others.

## IV. CLASSIC CRYPTOGRAPHY

The earliest forms of secret writing required little more than local pen and paper analogs, as most people could not read.

Fig 1: Reconstructed ancient Greek scytale (rhymes with "Italy"), an early cipher device

More literacy, or literate opponents, required actual cryptography. The main classical cipher types are transposition ciphers, which rearrange the order of letters in a message (e.g., 'hello world' becomes 'ehlol owrdl' in a trivially simple rearrangement scheme), and substitution ciphers, which systematically replace letters or groups of letters with other letters or groups of letters (e.g., 'fly at once' becomes 'gmz bu podf' by replacing each letter with the one following it in the Latin alphabet). Simple versions of either offered little confidentiality from enterprising opponents, and still do. An early substitution cipher was the Caesar cipher, in which each letter in the plaintext was replaced by a letter some fixed number of positions further down the alphabet. It was named after Julius Caesar who is reported to have used it, with a shift of 3, to communicate with his generals during his military campaigns, just like EXCESS-3 code in boolean algebra. There is record of several early Hebrew ciphers as well. The earliest known use of cryptography is some carved cipher text on stone in Egypt (ca 1900 BC), but this may have been done for the amusement of literate observers. The next oldest is bakery recipes from Mesopotamia.

Fig 2: 16th-century book-shaped French cipher machine, with arms of Henri II of France

Ciphertexts produced by a classical cipher (and some modern ciphers) always reveal statistical information about the plaintext, which can often be used to break them. After the discovery of frequency analysis perhaps by the Arab mathematician and polymath, Al-Kindi (also known as Alkindus), in the 9th century, nearly all such ciphers became more or less readily breakable by any informed attacker. Such classical ciphers still enjoy popularity today, though mostly as puzzles (see cryptogram). Al-Kindi wrote a book on cryptography entitled Risalah fi Istikhraj al-Mu'amma (Manuscript for the Deciphering Cryptographic Messages), in which described the first cryptanalysis techniques, including some for polyalphabetic ciphers.

## V. MODERN CRYPTOGRAPHY

**Symmetric-key cryptography:** Symmetric-key cryptography refers to encryption methods in which both the sender and receiver share the same key (or, less commonly, in which their keys are different, but related in an easily computable way). This was the only kind of encryption publicly known until June 1976.

The modern study of symmetric-key ciphers relates mainly to the study of block ciphers and stream ciphers and to their applications. A block cipher is, in a sense, a modern embodiment of Alberti's polyalphabetic cipher: block ciphers take as input a block of plaintext and a key, and output a block of ciphertext of the same size. Since messages are almost always longer than a single block, some method of knitting together successive blocks is required. Several have been developed, some with better security in one aspect or another than others. They are the modes of operation and must be carefully considered when using a block cipher in a cryptosystem.

The Data Encryption Standard (DES) and the Advanced Encryption Standard (AES) are block cipher designs which have been designated cryptography standards by the US government (though DES's designation was finally withdrawn after the AES was adopted). Despite its deprecation as an official standard, DES (especially its still-approved and much more secure triple-DES variant) remains quite popular; it is used across a wide range of applications, from ATM encryption to e-mail privacy and secure remote access. Many other block ciphers have been designed and released, with considerable variation in quality. Many have been thoroughly broken; see Category: Block ciphers.

Stream ciphers, in contrast to the 'block' type, create an arbitrarily long stream of key material, which is combined with the plaintext bit-by-bit or character-by-character, somewhat like the one-time pad. In a stream cipher, the output stream is created based on a hidden internal state which changes as the cipher operates. That internal state is initially set up using the secret key material. RC4 is a widely used stream cipher; Block ciphers can be used as stream ciphers;

Cryptographic hash functions are a third type of cryptographic algorithm. They take a message of any length as input, and output a short, fixed length hash which can be used in (for example) a digital signature. For good hash functions, an attacker cannot find two messages that produce the same hash. MD4 is a long-used hash function which is now broken; MD5, a strengthened variant of MD4, is also widely used but broken in practice. The U.S. National Security Agency developed the Secure Hash Algorithm series of MD5-like hash functions: SHA-0 was a flawed algorithm that the agency withdrew; SHA-1 is widely deployed and more secure than MD5, but cryptanalysts have identified attacks against it; the SHA-2 family improves on SHA-1, but it isn't yet widely deployed, and the U.S. standards authority thought it "prudent" from a security perspective to develop a new standard to "significantly improve the robustness of NIST's overall hash algorithm toolkit." Thus, a hash function design competition is underway and meant to select a new U.S. national standard, to be called SHA-3, by 2012.

Message authentication codes (MACs) are much like cryptographic hash functions, except that a secret key is used to authenticate the hash value on receipt.

**Public-key cryptography:** Symmetric-key cryptosystems use the same key for encryption and decryption of a message, though a message or group of messages may have a different key than others. A significant disadvantage of symmetric ciphers is the key management necessary to use them securely. Each distinct pair of communicating parties must, ideally, share a different key, and perhaps each ciphertext exchanged as well.

The number of keys required increases as the square of the number of network members, which very quickly requires complex key management schemes to keep them all straight and secret. The difficulty of securely establishing a secret key between two communicating parties, when a secure channel doesn't already exist between them, also presents a chicken-and-egg problem which is a considerable practical obstacle for cryptography users in the real world.

In a groundbreaking 1976 paper, Whitfield Diffie and Martin Hellman proposed the notion of *public-key* (also, more generally, called *asymmetric key*) cryptography in which two different but mathematically related keys are used—a *public* key and a *private* key. A public key system is so constructed that calculation of one key (the 'private key') is computationally infeasible from the other (the 'public key'), even though they are necessarily related. Instead, both keys are generated secretly, as an interrelated pair. The historian David Kahn described public-key cryptography as "the most revolutionary new concept in the field since polyalphabetic substitution emerged in the Renaissance".

In public-key cryptosystems, the public key may be freely distributed, while its paired private key must remain secret. The *public key* is typically used for encryption, while the *private* or *secret key* is used for decryption. Diffie and Hellman showed that public-key cryptography was possible by presenting the Diffie-Hellman key exchange protocol.

In 1978, Ronald Rivest, Adi Shamir, and Len Adleman invented RSA, another public-key system.

In 1997, it finally became publicly known that asymmetric key cryptography had been invented by James H. Ellis at GCHQ, a British intelligence organization, and that, in the early 1970s, both the Diffie-Hellman and RSA algorithms had been previously developed (by Malcolm J. Williamson and Clifford Cocks, respectively).

## VI. WHITE-BOX CRYPTOGRAPHY

In recent years, we have witnessed a trend towards the use of complex software applications with strong security requirements.

Think of banking applications, online games, and digital multimedia players. Prominent building blocks for these applications are cryptographic primitives, such as encryption schemes, digital signature schemes, and authentication mechanisms. Unfortunately, such building blocks (e.g., the AES encryption scheme) are guaranteed to be secure only when they are executed on a trustworthy system. It is known that several cryptographic primitives become insecure when the attacker has non-black-box (e.g., ‘whitebox’, or side-channel) access to the computation.

White-box cryptography (WBC) deals with protecting cryptographic primitives embedded in a program that the attacker has white-box access to. It aims to provide security when the program is executing in a hostile environment and the attacker can conduct non-black-box attacks (such as code inspection, execution environment modification, code modification, etc). However, no formal definitions of white-box cryptography were given, neither were there any proofs of security. With their subsequent cryptanalysis [3], it remains an open question whether or not such white-box implementations exist. In this paper, we initiate a study of rigorous security notions for the white-box setting.

One way to realize WBC is to obfuscate the executable code of the algorithm and hope that the adversary cannot use it in a non-black-box manner. What we would like is that an obfuscator ensures that all the security notions are satisfied in a white-box attack context when they are satisfied in the black-box attack context. However, it is still not clear if any existing definitions of obfuscation can be used to achieve this goal.

## VII. WHITE-BOX IMPLEMENTATIONS

The main objective of this implementation is to assess the security of cryptographic primitives in the presence of a white-box adversary. This research is denoted as *white-box cryptography*. In this we analyze the security of practical implementations of such primitives, denoted as *white-box implementations*. We refer to the methods and procedures of such an adversary as *white-box cryptanalysis*.

The main interest of white-box cryptography lies in the implementation of symmetric ciphers, in particular block ciphers.

We assess the security of these techniques, by analyzing the implementations and deploying cryptanalysis techniques. This is similar to the 'proof by scrutiny' approach. We also initiate an analysis of basic building blocks of block ciphers, and conclude with strategies towards secure white-box implementations by formulating 'white-box design criteria' for block ciphers.

**Attack Models:** One can distinguish three main attack models that capture the attack capabilities of an adversary on cryptosystems. These are the following:

**Black-box Model:** The black-box model is the traditional attack model, where an adversary has only access to the functionality of a cryptosystem.

**Grey-box Model** : The grey-box model refers to a model where a leakage function is present. In such an attack context, the adversary can deploy side-channel cryptanalysis techniques.

**White-box Model:** In the white-box model, the adversary has total visibility of the software implementation of the cryptosystem, and full control over its execution platform. One could refer to the white-box model as the worst-case model, where in contrast to grey-box models, it is impossible for an adversary not to comply with the this model. The white-box model is used to analyze algorithms that are running in an untrusted environment, that is, an environment in which applications are subject to attacks from the execution platform.

## VIII. OBJECTIVES OF WBC

White-Box Cryptography, when it was proposed in 2001 by Chow et al. [43], was originally defined as an obfuscation technique with the following objective.

**Def.1 :** White-Box Cryptography is an obfuscation technique intended to implement cryptographic primitives in such a way, that even an adversary who has full access to the implementation and its execution platform, is unable to extract key information.

**Def.2:** The objective of White-Box Cryptography is to implement cryptographic primitives in such a way that, within the context of the intended application, having full access to the cryptographic implementation does not present any advantage for a computationally bounded adversary in comparison to the adversary dealing with the implementation as a black box.

## IX. OBFUSCATION STRATEGY

Code obfuscation is the most viable method to prevent reverse-engineering [4].A code obfuscator is used to convert a code (program) into an equivalent one that is difficult to reverse engineer, by distinguishing its internal workings. We denote an obfuscator as $O$, and the obfuscation of the program $P$ as $O(P)$. The first contributions towards a formalization of code obfuscation were made by Hada [5], who presented definitions for obfuscation based on the *simulation paradigm* for *zero knowledge*, called GMR-ZK, given in [6]. Using simulation is an approach that has been used before in formal security proofs for cryptography.

In this approach, there are two settings. One setting in which an arbitrary, probabilistic, polynomial-time adversary can interact with the cryptographic primitive; in the other setting, the adversary interacts with an idealized version of the cryptographic primitive that can never be broken. This idealized version is an abstract primitive. E.g., a perfectly secure encryption functions as an idealized abstract version of a practical encryption scheme. To determine whether a primitive is secure, the output of the adversaries in the two settings is compared. If their outputs are approximately the same (or indistinguishable in the case of output distributions), then the cryptographic primitive must be secure, since the idealized version is secure.

The main difference between the obfuscation definition and the simulation based definitions used in (black-box) cryptography is in the type of objects the adversary interacts with. In the obfuscation case, it is a comparison between (white-box) interaction to an implementation of the primitive, and the interaction with a oracle implementation (black-box) [5]. In the tradition cryptography case, it is between an oracle implementation of the cryptographic primitive, and an idealized version. This new concept is captured by the *Virtual Black-Box Property (VBBP)*.

## REFERENCES

[1] R.Rivest "The MD5 Message Digest Algorithm," RFC 1321, Network Working Group Request for Comments, April 1992
[2] FIPS. 46-3, "Data Encryption Standard," Federal Information Processing Standard (FIPS), Publication 180-2, National Bureau of Standards, US, August 2002
[3] O. Billet, H. Gilbert, C. Ech-Chatbi, *Cryptanalysis of a White- box AES Implementation*, SAC 2004.
[4] Christian Collberg, Clark Thomborson, and Douglas Low. A Taxonomy of Obfuscating Transformations. Technical Report 148, July 1997.
[5] Satoshi Hada. Zero-Knowledge and Code Obfuscation. In Tatsuaki Okamoto, editor, *Advances in Cryptology Science*, *ASIACRYPT 2000*, volume 1976 of *Lecture Notes in Computer* pages 443–457, London, UK, 2000. Springer-Verlag.

[6] S Goldwasser, S Micali, and C Rackoff. The knowledge complexity of interactive proof-systems. In *Proceedings of the 17th annual ACM Symposium on Theory of Computing (STOC 1985)*, pages 291–304, New York,

[7] Boaz Barak. How to Go Beyond the Black-Box Simulation Barrier. In *Proceedings of the 42nd symposium on Foundations of Computer Science (FOCS 2001)*, IEEE Computer Society, pages106–115, Washington, DC, USA, 2001. IEEE Computer