УДК 004.056

ПРОБЛЕМИ ЗАХИСТУ ФАЙЛІВ ВІД НЕСАНКЦІОНОВАНОГО КОПІЮВАННЯ.

Ешмідт А.К.
Науковий керівник – проф., д.т.н. Яремчук Ю.Є

На даному етапі розвитку інформаційних технологій надзвичайно актуальним є питання якісного захисту даних від несанкціонованого копіювання. Кожна організація або окремий користувач володіють конфіденційними даними, або документами з персональною і не призначеною для публічного доступу інформацією. У таких випадках необхідно використовувати спеціальні ПЗ, що захищатимуть таку інформацію від несанкціонованого копіювання. Найпоширенішим засобом захисту є шифрування файлу за допомогою програми, тобто перетворення його у нечитабельний вигляд. Отже, якщо зловмисник скопіює захищений файл, то не зможе його прочитати без ключа розшифрування. Метою даного дослідження є розробка алгоритму захисту, який надалі можна буде використати під час розроблення ПЗ для захисту файлів від несанкціонованого копіювання. Пропонується використовувати алгоритм шифрування інформації методом гамування. Для формування гами пропонується використовувати системні файли операційної системи, завдяки чому відбуватиметься прив'язка захищеного файлу до операційної системи. Для підвищення стійкості шифрування пропонується використовувати гаму як результат замішування змісту двох або більшої кількості системних файлів. Пропонується використовувати два файли $G_1=\{g_{11}, g_{12},\ldots, g_{1n}\}$ та $G_2=\{g_{21}, g_{22},\ldots, g_{2n}\}$, які формують гаму, та файл $G_3=\{g_{31}, g_{32},\ldots, g_{3n}\}$, який керує процесом формування гами. Залежно від того, який біт керуючого файлу G_3 буде поступати на логічний блок, 1 або 0, буде обраний відповідний біт з потоку G_1 чи G_2. Тобто, якщо зчитаний біт дорівнює нулю, то із потоку G_1 зчитується один біт, який накладається на один біт вхідної інформації як гама. Якщо ж зчитаний біт дорівнює одиниці, то із потоку G_2 зчитується один біт, який накладається на один біт вхідної інформації як гама. Стійкість гами можна підвищити за рахунок зчитування двох бітів керуючого файлу G_3 замість одного. У цьому випадку керування двома бітами виявляється у тому, що перший біт керує вибором біту з файлу G_1, а другий біт – G_2. При цьому значення біту 0 означає, що береться пряме значення біту, а значення біту 1 – інверсне значення біту. Для найбільш оптимального вибору методу проведено аналіз стійкості шифрування.

Отже, запропоновано алгоритм шифрування файлів за допомогою гамування зі створенням гами на основі системних файлів ОС. Результати даної роботи можуть бути застосовані під час розроблення ПЗ для захисту файлів від несанкціонованого копіювання.