StegFS

Steganograficzny system plików

Michał Politowski

mp169814@students.mimuw.edu.pl

Kryptograficzne systemy plików

Kryptograficzne systemy plików (np. TCFS) i mechanizmy szyfrowania urządzeń dyskowych (np. oferowane przez urządzenia typu loop w Linuksie):

- Pozwalają przechowywać dane w postaci zaszyfrowanej.

Kryptograficzne systemy plików

Kryptograficzne systemy plików (np. TCFS) i mechanizmy szyfrowania urządzeń dyskowych (np. oferowane przez urządzenia typu loop w Linuksie):

- Pozwalają przechowywać dane w postaci zaszyfrowanej.
- Zapewniają bezpieczeństwo danych gdy dysk z nimi dostanie się w niepowołane ręce. Kryptoanaliza w przypadku porawnie zaimplementowanych skutecznych szyfrów jest bardzo kosztowna, często nieopłacalna.

Kryptograficzne systemy plików

Kryptograficzne systemy plików (np. TCFS) i mechanizmy szyfrowania urządzeń dyskowych (np. oferowane przez urządzenia typu loop w Linuksie):

- Pozwalają przechowywać dane w postaci zaszyfrowanej.
- Zapewniają bezpieczeństwo danych gdy dysk z nimi dostanie się w niepowołane ręce. Kryptoanaliza w przypadku porawnie zaimplementowanych skutecznych szyfrów jest bardzo kosztowna, często nieopłacalna.
- *Nie zabezpieczają* przed wymuszeniem ujawnienia kluczy deszyfrujących przez osoby lub instytucje, które wiedzą o istnieniu zaszyfrowanych danych.

Steganografia

- *Steganografia* – określenie technik mających na celu ukrycie samego istnienia danych.

Steganografia

- *Steganografia* – określenie technik mających na celu ukrycie samego istnienia danych.
- Popularną metodą jest zapisywanie danych do ukrycia jako modyfikacji najmniej znaczących bitów danych graficznych lub dźwiękowych.

Steganografia

- *Steganografia* – określenie technik mających na celu ukrycie samego istnienia danych.
- Popularną metodą jest zapisywanie danych do ukrycia jako modyfikacji najmniej znaczących bitów danych graficznych lub dźwiękowych.
- Najbardziej chyba znanym programem tego typu jest OutGuess, można też w ten sposób użyć linuxowego urządzenia loop.

Steganografia

- *Steganografia* – określenie technik mających na celu ukrycie samego istnienia danych.
- Popularną metodą jest zapisywanie danych do ukrycia jako modyfikacji najmniej znaczących bitów danych graficznych lub dźwiękowych.
- Najbardziej chyba znanym programem tego typu jest OutGuess, można też w ten sposób użyć linuxowego urządzenia loop.
- StegFS jest opartym o Ext2FS systemem plików zapewniającym *wiarygodną zaprzeczalność* (plausible deniability) istnienia zaszyfrowanych danych.

Steganografia

- *Steganografia* – określenie technik mających na celu ukrycie samego istnienia danych.
- Popularną metodą jest zapisywanie danych do ukrycia jako modyfikacji najmniej znaczących bitów danych graficznych lub dźwiękowych.
- Najbardziej chyba znanym programem tego typu jest OutGuess, można też w ten sposób użyć linuxowego urządzenia loop.
- StegFS jest opartym o Ext2FS systemem plików zapewniającym *wiarygodną zaprzeczalność* (plausible deniability) istnienia zaszyfrowanych danych.
- W założeniu przeciwnik, wiedzący oczywiście o istnieniu na dysku systemu plików StegFS, nie może uzyskać pewności, że ujawniono klucze deszyfrujące dla wszystkich zapisanych w nim danych.

Ukrywanie danych na urządzeniu blokowym

Dla uniemożliwienia określenia czy na dysku znajdują się jakieś dane i ile ich tam jest można postąpić następująco:

- Dysk wypełniamy początkowo w całości losowo.
- Dane ukrywane są przez zapisywanie po zaszyfrowaniu w blokach pseudolosowo wybranych na podstawie hasła i nazwy pliku.
  - Nie można dzięki temu odróżnić bloków zajętych przez dane od wolnych.

Ukrywanie danych na urządzeniu blokowym

- Problemem tej metody są kolizje powodujące nadpisywanie zajętych bloków.
- Kolizje wystąpią z prawdopodobieństwem $\geq \frac{1}{2}$ już przy wykorzystaniu $\sqrt{n}$ bloków, gdzie n jest liczbą bloków na dysku (paradoks dnia urodzin).
- Rozwiązaniem jest przechowywanie na dysku wielu kopii każdego bloku danych, konieczna jest też metoda wykrywania przy odczycie, że dany blok został nadpisany.
  - Nawet to nie gwarantuje jednak oczywiście pełnego bezpieczeństwa danych.

StegFS – dwa systemy plików w jednym

StegFS korzysta z przedstawionej metody z pewnymi modyfikacjami:

StegFS – dwa systemy plików w jednym

StegFS korzysta z przedstawionej metody z pewnymi modyfikacjami:

- Ukryte pliki zapisywane są w blokach wolnych z punktu widzenia istniejącego na tej samej partycji standardowego systemu plików ext2.

StegFS – dwa systemy plików w jednym

StegFS korzysta z przedstawionej metody z pewnymi modyfikacjami:

- Ukryte pliki zapisywane są w blokach wolnych z punktu widzenia istniejącego na tej samej partycji standardowego systemu plików ext2.
- Każdy plik ukryty przydzielony jest do jednego z 15 *poziomów bezpieczeństwa*, które można niezależnie ujawniać i ukrywać.

StegFS – dwa systemy plików w jednym

StegFS korzysta z przedstawionej metody z pewnymi modyfikacjami:

- Ukryte pliki zapisywane są w blokach wolnych z punktu widzenia istniejącego na tej samej partycji standardowego systemu plików ext2.
- Każdy plik ukryty przydzielony jest do jednego z 15 *poziomów bezpieczeństwa*, które można niezależnie ujawniać i ukrywać.
- Zamiast funkcji mieszającej nazwę pliku i hasło wykorzystywana jest dodatkowa *tablica alokacji bloków* – odpowiednik mapy bitowej z np. ext2.

StegFS – dwa systemy plików w jednym

StegFS korzysta z przedstawionej metody z pewnymi modyfikacjami:

- Tablica alokacji bloków zawiera dla każdego bloku 128-bitowy zaszyfrowany wpis. Jej rozmiar jest stały, zależny tylko od rozmiaru używanego urządzenia blokowego, nie od wykorzystywanych poziomów bezpieczeństwa StegFS.

StegFS – dwa systemy plików w jednym

StegFS korzysta z przedstawionej metody z pewnymi modyfikacjami:

- Tablica alokacji bloków zawiera dla każdego bloku 128-bitowy zaszyfrowany wpis. Jej rozmiar jest stały, zależny tylko od rozmiaru używanego urządzenia blokowego, nie od wykorzystywanych poziomów bezpieczeństwa StegFS.
- Jawna część StegFS jest kompatybilna z systemem plików ext2. Można wykorzystywać implementację ext2 do jej obsługi, a także używać StegFS do obsługi systemów plików ext2.
  - Pozwala to na całkowite usunięcie obsługi StegFS z systemu z zachowaniem dostępu do jawnej części danych.

StegFS – ext2

Istnieje kilka różnic między implementacją ext2 w StegFS a standardową:

- Zwalniane bloki są nadpisywane losowymi danymi
  - nie można odróżnić bloków wolnych od wykorzystywanych przez ukryte pliki.

StegFS – ext2

Istnieje kilka różnic między implementacją ext2 w StegFS a standardową:

- Zwalniane bloki są nadpisywane losowymi danymi
  - nie można odróżnić bloków wolnych od wykorzystywanych przez ukryte pliki.
- Niewielki procent alokacji bloków jest przeprowadzany losowo
  - nie można stwierdzić, czy zmienione między inspekcjami wolne bloki są wykorzystywane przez ukryte pliki, czy były wykorzystywane przez krótko istniejący plik jawny.

StegFS – ext2

Istnieje kilka różnic między implementacją ext2 w StegFS a standardową:

- Zwalniane bloki są nadpisywane losowymi danymi
  - nie można odróżnić bloków wolnych od wykorzystywanych przez ukryte pliki.
- Niewielki procent alokacji bloków jest przeprowadzany losowo
  - nie można stwierdzić, czy zmienione między inspekcjami wolne bloki są wykorzystywane przez ukryte pliki, czy były wykorzystywane przez krótko istniejący plik jawny.
- Alokacje bloków dla plików jawnych uwzględniają oczywiście także położenie plików ukrytych z aktualnie dostępnych poziomów bezpieczeństwa.

Tablica alokacji bloków

- Stanowi odpowiednik mapy bitowej alokacji bloków.

Tablica alokacji bloków

- Stanowi odpowiednik mapy bitowej alokacji bloków.
- Przechowywana jest w pliku jawnym.

Tablica alokacji bloków

- Stanowi odpowiednik mapy bitowej alokacji bloków.
- Przechowywana jest w pliku jawnym.
- Zawiera zaszyfrowany wpis długości 128 bitów dla każdego bloku na dysku.

Tablica alokacji bloków

- Stanowi odpowiednik mapy bitowej alokacji bloków.
- Przechowywana jest w pliku jawnym.
- Zawiera zaszyfrowany wpis długości 128 bitów dla każdego bloku na dysku.
- Do szyfrowania używany jest ten sam klucz, którym zaszyfrowany jest dany blok.

Tablica alokacji bloków

- Stanowi odpowiednik mapy bitowej alokacji bloków.
- Przechowywana jest w pliku jawnym.
- Zawiera zaszyfrowany wpis długości 128 bitów dla każdego bloku na dysku.
- Do szyfrowania używany jest ten sam klucz, którym zaszyfrowany jest dany blok.
- Wpisy dla nieużywanych bloków wypełniane są losowymi danymi.

Tablica alokacji bloków

- Stanowi odpowiednik mapy bitowej alokacji bloków.
- Przechowywana jest w pliku jawnym.
- Zawiera zaszyfrowany wpis długości 128 bitów dla każdego bloku na dysku.
- Do szyfrowania używany jest ten sam klucz, którym zaszyfrowany jest dany blok.
- Wpisy dla nieużywanych bloków wypełniane są losowymi danymi.
- Plik tablicy na końcu zawiera dodatkowo macierz bezpieczeństwa (opisaną dalej).

Tablica alokacji bloków

```
struct stegfs_btable {
    uint32_t magic1;
    uint16_t magic2;
    uint16_t iv;
    uint32_t bchecksum;
    uint32_t ino;
}
```

`magic1` równe 1 dla poprawnych wpisów,

`magic2` równe 0 dla poprawnych wpisów bloków danych,
1 – dla poprawnych wpisów i-węzłów,

`iv` wektor startowy szyfru,

`bchecksum` suma kontrolna do wykrywania nadpisanych bloków – ostatnie 32 bity bloku,

`ino` numer i-węzła zapisanego w danym bloku.

I-węzły

I-węzeł na ukrytym poziomie systemu StegFS jest wzorowany na i-węźle systemu ext2, zawiera jednak dodatkowe informacje.

```
struct stegfs_inode {
    ...  /* Jak w ext2 oprócz i_block */
    __u8 i_icopies;
    __u8 i_bcopies;
    __u16 i_pad0;
    __u32 i_inode[STEGFS_MAX_INO_COPIES];
    __u32 i_block[EXT2_N_BLOCKS *
                  STEGFS_MAX_BLOCK_COPIES];
};
```

I-węzły

`i_icopies` liczba kopii i-węzła,

`i_bcopies` liczba kopii bloków danych,

`i_inode` numery bloków kopii i-węzła,

`i_block` numery wszystkich kopii bloków danych,

`EXT2_N_BLOCKS` 15 – tak jak w ext2 mamy 12 bloków bezpośrednich i po jednym pośrednim, podwójnie pośrednim i potrójnie pośrednim.

`STEGFS_MAX_INO_COPIES` 28 – maksymalna liczba kopii i-węzła,

`STEGFS_MAX_BLOCK_COPIES` 14 – maksymalna liczba kopii bloku danych.

I-węzły

- Rozmiar ukrytego i-węzła wynosi dokładnie 1024 bajty. Każdy ukryty i-węzeł zajmuje zawsze jeden blok, nawet gdy bloki są większe.

I-węzły

- Rozmiar ukrytego i-węzła wynosi dokładnie 1024 bajty. Każdy ukryty i-węzeł zajmuje zawsze jeden blok, nawet gdy bloki są większe.
- Liczba kopii i-węzła i danych nowo tworzonego pliku jest dziedziczona z katalogu macierzystego lecz można ją zmienić (wywołaniem ioctl).
  - Im częściej jest wykorzystywany dany poziom bezpieczeństwa tym mniej kopii potrzeba dla zabezpieczenia przed zniszczeniem danych.

I-węzły

Numery są przydzielane i-węzłom w zależności od poziomu bezpieczeństwa, na którym są tworzone:

- Dla i-węzłów jawnych:

31	30	29 … 0
0	0	reszta numeru

I-węzły

Numery są przydzielane i-węzłom w zależności od poziomu bezpieczeństwa, na którym są tworzone:

- Dla i-węzłów jawnych:

31	30	29 . . . 0
0	0	reszta numeru

- Dla i-węzłów ukrytych:

31	30	29 . . . 26	25 . . . 0
0	1	poziom	reszta numeru

Przydział bloków dla danych ukrytych

Każdy blok urządzenia może być użyty przez jawną część systemu plików, nie można więc przydzielić stałych lokacji na urządzeniu żadnym danym ani metadanym ukrytych poziomów bezpieczeństwa, łącznie z i-węzłami ich katalogów głównych.

- Wybór bloków dla i-węzła
  - Aby nie przeszukiwać całej tablicy bloków w poszukiwaniu i-węzła o danym numerze numer bloku wyznacza się korzystając z funkcji mieszającej na podstawie klucza dla danego poziomu bezpieczeństwa, numeru i-węzła i numeru kolejnego, zwiększanego aż do znalezienia odpowiedniej liczby wolnych bloków.

Przydział bloków dla danych ukrytych

Każdy blok urządzenia może być użyty przez jawną część systemu plików, nie można więc przydzielić stałych lokacji na urządzeniu żadnym danym ani metadanym ukrytych poziomów bezpieczeństwa, łącznie z i-węzłami ich katalogów głównych.

- Wybór bloków dla danych
  - Bloki danych i bloki pośrednie ukrytych plików rozmieszczane są losowo.
  - Dla każdej kopii alokowane są niezależnie kolejne wolne bloki zaczynając od losowo wybranego. Niezależność alokacji ma zapewnić niezależność prawdopodobieństw nadpisania.

Przydział bloków dla danych ukrytych

Każdy blok urządzenia może być użyty przez jawną część systemu plików, nie można więc przydzielić stałych lokacji na urządzeniu żadnym danym ani metadanym ukrytych poziomów bezpieczeństwa, łącznie z i-węzłami ich katalogów głównych.

- Wybór bloków dla danych
  - **Uwaga**: losowanie położenia każdego bloku pogorszyłoby wydajność i bezpieczeństwo:
    - Jawne pliki tworzone przy otwartych ukrytych poziomach miałyby częste przerwy rozmiaru bloku;
    - ich duża liczba mogłaby ułatwić oszacowanie ilości ukrytych danych na dysku.
    - Wzory alokacji bloków plików ukrytych i jawnych muszą być więc podobne.

Przydział bloków – wolne bloki

Blok jest uważany za wolny jeśli:

- Jest oznaczony jako wolny w mapie bitowej dla plików jawnych,
- i pierwsze 47 bitów jego wpisu w tablicy alokacji nie jest zerami po odszyfrowaniu żadnym z aktualnie dostępnych kluczy poziomów bezpieczeństwa.

Zwalnianie bloków

- Przy zwalnianiu bloku jest on, jak i jego wpis w tablicy bloków, zapisywany losowymi danymi.
- Dotyczy to także bloków plików jawnych – skasowane pliki jawne muszą być nieodróżnialne od plików ukrytych bez dostępnych kluczy.

Powielanie bloków

- Ukryte i-węzły i dane są zapisywane na dysku w wielu kopiach aby zapobiec ich utracie przy nadpisaniu.

Powielanie bloków

- Ukryte i-węzły i dane są zapisywane na dysku w wielu kopiach aby zapobiec ich utracie przy nadpisaniu.
- Przy odczycie sprawdzamy sumę kontrolną i deszyfrujemy pierwszy blok z poprawną sumą.

Powielanie bloków

- Ukryte i-węzły i dane są zapisywane na dysku w wielu kopiach aby zapobiec ich utracie przy nadpisaniu.
- Przy odczycie sprawdzamy sumę kontrolną i deszyfrujemy pierwszy blok z poprawną sumą.
- Przy zapisie należy uaktualnić wszystkie kopie bloku oraz sumę kontrolną w tablicy alokacji bloków.

Powielanie bloków

- Ukryte i-węzły i dane są zapisywane na dysku w wielu kopiach aby zapobiec ich utracie przy nadpisaniu.
- Przy odczycie sprawdzamy sumę kontrolną i deszyfrujemy pierwszy blok z poprawną sumą.
- Przy zapisie należy uaktualnić wszystkie kopie bloku oraz sumę kontrolną w tablicy alokacji bloków.
- Jeśli któryś blok został w międzyczasie zajęty przez plik, o którego istnieniu wiemy, należy zaalokować nowy blok uaktualniając i-węzeł oraz tablicę alokacji bloków.

Powielanie bloków

- Ukryte i-węzły i dane są zapisywane na dysku w wielu kopiach aby zapobiec ich utracie przy nadpisaniu.
- Przy odczycie sprawdzamy sumę kontrolną i deszyfrujemy pierwszy blok z poprawną sumą.
- Przy zapisie należy uaktualnić wszystkie kopie bloku oraz sumę kontrolną w tablicy alokacji bloków.
- Jeśli któryś blok został w międzyczasie zajęty przez plik, o którego istnieniu wiemy, należy zaalokować nowy blok uaktualniając i-węzeł oraz tablicę alokacji bloków.

Wniosek: do odtworzenia pełnej liczby kopii wszystkich danych pliku wystarczy jego odczyt i ponowny zapis. Ze względu na uaktualnianie czasu ostatniego dostępu i-węzły są zapisywane, więc także odtwarzane, przy każdym dostępie do pliku.

Zarządzanie kluczami i szyfrowanie

- Każdy ukryty plik należy do dokładnie jednego poziomu bezpieczeństwa z dostępnych 15.

Zarządzanie kluczami i szyfrowanie

- Każdy ukryty plik należy do dokładnie jednego poziomu bezpieczeństwa z dostępnych 15.
- Każdy poziom bezpieczeństwa należy do dowolnych z 15 dostępnych kontekstów bezpieczeństwa.

Zarządzanie kluczami i szyfrowanie

- Każdy ukryty plik należy do dokładnie jednego poziomu bezpieczeństwa z dostępnych 15.
- Każdy poziom bezpieczeństwa należy do dowolnych z 15 dostępnych kontekstów bezpieczeństwa.
- Macierz bezpieczeństwa zapisana na końcu pliku tablicy alokacji bloków zawiera 15×15 128-bitowych wpisów.

Zarządzanie kluczami i szyfrowanie

- Każdy ukryty plik należy do dokładnie jednego poziomu bezpieczeństwa z dostępnych 15.
- Każdy poziom bezpieczeństwa należy do dowolnych z 15 dostępnych kontekstów bezpieczeństwa.
- Macierz bezpieczeństwa zapisana na końcu pliku tablicy alokacji bloków zawiera 15×15 128-bitowych wpisów.
- Dla każdego kontekstu bezpieczeństwa można i należy używać innego hasła.

Zarządzanie kluczami i szyfrowanie

- Każdy ukryty plik należy do dokładnie jednego poziomu bezpieczeństwa z dostępnych 15.
- Każdy poziom bezpieczeństwa należy do dowolnych z 15 dostępnych kontekstów bezpieczeństwa.
- Macierz bezpieczeństwa zapisana na końcu pliku tablicy alokacji bloków zawiera 15×15 128-bitowych wpisów.
- Dla każdego kontekstu bezpieczeństwa można i należy używać innego hasła.
- Domyślnie kontekst C obejmuje poziomy $1, \ldots, C$.

Zarządzanie kluczami i szyfrowanie

- Z hasła kontekstu PP_C używając kryptograficznej funkcji skrótu uzyskuje się klucz kontekstu HP_C

$$HP_C = h(PP_C)$$

Zarządzanie kluczami i szyfrowanie

- Z hasła kontekstu PP_C używając kryptograficznej funkcji skrótu uzyskuje się klucz kontekstu HP_C

$$HP_C = h(PP_C)$$

- Jeśli poziom bezpieczeństwa $L \in C$ to odpowiedni wpis w macierzy bezpieczeństwa zawiera zaszyfrowany losowy klucz poziomu SK_L.

$$M_{C,L} = E_{HP_C}(SK_L)$$

W przeciwnym przypadku wpis w macierzy bezpieczeństwa zawiera losowe dane.

Zarządzanie kluczami i szyfrowanie

- Blok i należący do pliku z poziomu bezpieczeństwa L, oraz jego wpis w tablicy alokacji bloków, szyfrowane są kluczem $BK_{L,i}$, uzyskiwanym z klucza poziomu i numeru bloku operacją XOR.

$$BK_{L,i} = SK_L \oplus i$$

Użycie StegFS

- Utworzenie systemu plików:
  - `mke2fs /dev/urządzenie`
  - `mkstegfs /dev/urządzenie /ścieżka/do/tablicy/alokacji`
    - Podajemy hasła do wszystkich 15 kontekstów bezpieczeństwa.
- Montowanie:
  `mount /dev/urządzenie /punkt/montowania -t stegfs -o btab=/ścieżka/do/tablicy/alokacji`

Użycie StegFS

- Manipulacja kontekstami bezpieczeństwa:
  - `stegfsopen /punkt/montowania kontekst`
    - pojawiają się katalogi `/punkt/montowania/stegfs/n` dla poziomów `n` należących do `kontekstu`
  - `stegfsclose /punkt/montowania poziom`
  - `stegfsctrl add kontekst poziom`
  - `stegfsctrl remove kontekst poziom`
- Odtwarzanie powielania: `rerepl plik`
  - Odczytuje i zapisuje ponownie plik

Wydajność

Testy przeprwadzone prze autorów StegFS.

- Program testujący: Bonnie.
- Komputer: AMD K5 PR150 100 MHz, 1 GB partycja na 1.2 GB dysku Fujitsu.
- Współczynnik powielania: 5 (dla i-węzłów i danych).

	Zapis sekwencyjny		
	znakami (putc) [kB/s]	blokami (write) [kB/s]	nadpisanie (read/lseek/write) [kB/s]
Ext2	1835	3839	1964
Pliki jawne	1628	2663	1761
Pliki ukryte	44	45	10

Wydajność

Testy przeprwadzone prze autorów StegFS.

- Program testujący: Bonnie.
- Komputer: AMD K5 PR150 100 MHz, 1 GB partycja na 1.2 GB dysku Fujitsu.
- Współczynnik powielania: 5 (dla i-węzłów i danych).

	Odczyt sekwencyjny	
	znakami (getc) [kB/s]	blokami (read) [kB/s]
Ext2	2216	5476
Pliki jawne	2075	4872
Pliki ukryte	374	420

Wydajność

Testy przeprwadzone prze autorów StegFS.

- Program testujący: Bonnie.
- Komputer: AMD K5 PR150 100 MHz, 1 GB partycja na 1.2 GB dysku Fujitsu.
- Współczynnik powielania: 5 (dla i-węzłów i danych).
- Dla plików jawnych wydajność jest porównywalna z systemem ext2.
- Dla plików ukrytych wydajność jest znacznie gorsza: konieczność powielania bloków, koszty szyfrowania, konieczność wykrywania nadpisanych bloków.

Podsumowanie

Podstawowe cechy StegFS:

- poufnośc danych dzięki szyfrowaniu,
- wiarygodnie zaprzeczalne ukrywanie danych,
- możliwość ujawnienia niektórych, mniej istotnych danych w celu uzasadnienia użycia systemu,
- niszczenie kasowanych danych jawnych i ukrytych,
- zapis do ukrytych plików między inspekcjami przeciwnika nieodróżnialny od utworzenia i usunięcia plików jawnych,
- dostępność jawnych plików bez zainstalowanej implementacji StegFS,

Podsumowanie

Podstawowe cechy StegFS:

- Wielokrotne dowiązania do plików dopuszczalne tylko w obrębie jednego poziomu bezpieczeństwa,
- możliwe przypadkowe zniszczenie ukrytych plików – replikacja danych ogranicza niebezpieczeństwo,
- pełne bezpieczeństwo plików gdy wszystkie poziomy są otwarte,
- niska wydajność dostępu do ukrytych plików – replikacja, szyfrowanie, kontrola nadpisania.

Alternatywne rozwiązanie:

- Rubberhose – steganograficzne urządzenie blokowe, można na nim założyć dowolny system plików.