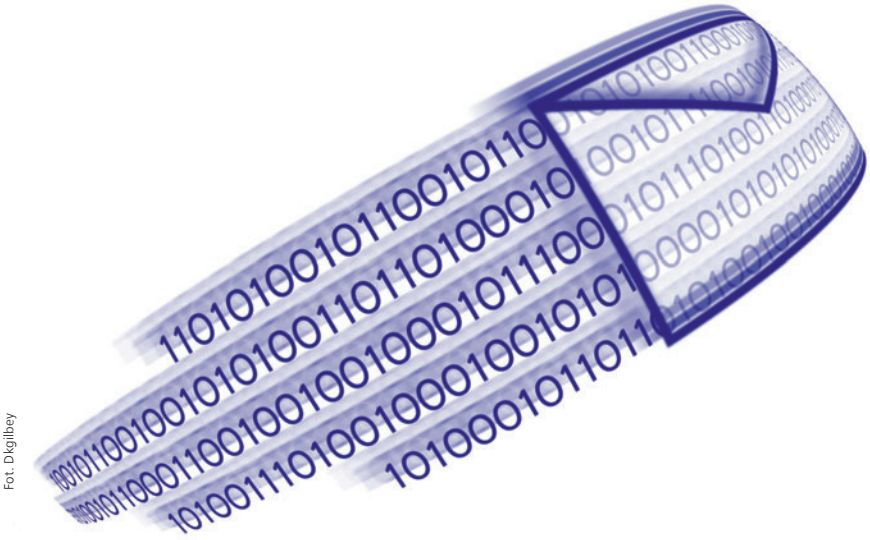
Zbigniew Gołębiewski,
Mirosław Kutyłowski, Filip Zagórski

Ataki kleptograficzne – ciemna strona kryptografii



Fot. Dkgilbey

Kryptografia stała się podstawowym narzędziem, zapewniającym bezpieczną wymianę danych w środowisku TCP/IP. Bez protokołów takich jak SSL, trudno wyobrazić sobie większość zastosowań komercyjnej wymiany danych w takich systemach. Rozwiązania oparte na kryptografii są tanią i efektywną alternatywą w stosunku do klasycznych, „fizycznych" metod ochrony. Okazuje się jednak, że również dla kryptografii istnieje „ciemna strona mocy"– metody kryptograficzne mogą być wykorzystane w perfidny sposób przeciwko jego użytkownikowi. Jedną z najbardziej wyrafinowanych metod jest tzw. kleptografia wynaleziona przez Adama Younga z Microsoft Research i Moti Yunga z Columbia University.

Coraz większego znaczenia nabiera problem ochrony danych przechowywanych w systemie komputerowym firmy czy instytucji przed nieuprawnionym przekazaniem na zewnątrz. Dostępność protokołów komunikacyjnych, liczba osób mających dostęp do danych, wreszcie możliwość samodzielnego działania przez szpiegowskie oprogramowanie – wszystko to sprawia, że problem nie jest łatwy do rozwiązania.

Możliwość nieuprawnionego przekazywania danych przez pracowników, ogranicza się za pomocą odpowiednich postanowień polityki bezpieczeństwa. Redukuje się możliwość korzystania z protokołów SSH i SCP, dla dostępu do komputerów leżących poza siecią korporacyjną, używania programów szyfrujących i zamieszczania szyfrowanych plików jako załączników w poczcie elektronicznej, stosowanie centralnych serwerów pocztowych dla korporacji z automatyczną kontrolą przesyłanych zawartości.

Kolejnym problemem jest wysyłanie informacji przez wrogie oprogramowanie, które może przedostać się do komputerów sieci korporacyjnej. Oprócz ochrony przed dostaniem się takiego oprogramowania do sieci korporacyj-

nej, monitorowane mogą być komunikaty wysyłane na zewnątrz.

Kleptografia

Czym jest kleptografia? To wyjątkowo perfidny rodzaj ataku atakujący narzędzia do legalnego transferu danych. Zasadniczymi cechami kleptografii są:

- wyciek informacji (w tym tajnych kluczy) za pomocą pakietów danych wysyłanych zgodnie ze specyfikacją protokołów kryptograficznych, w sposób nieodróżnialny od prawidłowego wykonania owych protokołów
- możliwość odkodowania owych informacji, **jedynie przez osobę posiadającą określone tajne klucze kleptograficzne.**

Od techniki kanału podprogowego, kleptografię odróżnia fakt, że klucze kleptograficzne, nie znajdują się w zarażonym programie. Tak więc *reverse engineering*, może co najwyżej wykazać obecność ataku kleptograficznego, niemożliwe pozostanie jednak przeprowadzenie na tej podstawie ataku na innych użytkowników, posługujących się programami zarażonymi w ten sam sposób.
W jaki sposób kleptografia może się rozprzestrzeniać? Zasadniczo są dwie drogi: jedna to zainstalowanie odpowiednio napisanego oprogramowania dostarczanego przez producenta w skompilowanej postaci. W związku z naturą kleptografii, testy dokonywane pod kątem zgodności produktu z zakładanym protokołem, nie wykażą w takim wypadku niczego podejrzanego. Druga droga, to wykorzystanie wirusów. Wirus taki będzie stosunkowo łatwy do napisania, bowiem infekowany ma być znany program (np. przeglądarka internetowa), można więc ograniczyć prawdopodobieństwo wystąpienia efektów ubocznych, pozwalających na zaobserwowanie nieprawidłowego zachowania się systemu, z punktu widzenia użytkownika.

Szczegóły techniczne

Przybliżmy Czytelnikowi podstawowy mechanizm ataku kleptograficznego, na przykładzie protokołu uzgadniania kluczy Diffiego-Hellmana. Przypomnijmy, że aby uzgodnić tajny klucz sesyjny, Alicja i Bob wykonują następujące kroki:

1. Alicja wybiera losowo liczbę a, następnie oblicza $A:=g^a \bmod p$. (liczby g oraz p są odpowiednimi parametrami, pomińmy nieistotne dla naszego opisu szczegóły dotyczące wyboru tych parametrów).
2. Alicja wysyła A do Boba.
3. Bob wybiera losowo liczbę b, następnie oblicza $B:=g^b \bmod p$.
4. Bob wysyła B do Alicji.

W tym momencie Alicja oblicza klucz sesyjny jako:

$$K=B^a \bmod p$$

Z kolei Bob oblicza:

$$K=A^b \bmod p$$

Łatwo zauważyć, że w obu przypadkach klucz K to

$$g^{ab} \bmod p$$

Genialność tej metody polega na fakcie, że mając liczby A i B nie sposób obliczyć K bez zlogarytmowania A lub B, a to jest praktycznie niewykonalne dla użytego p.

Na czym polega atak kleptograficzny przeciwko opisanemu protokołowi? Otóż załóżmy, że komputer Alicji został zainfekowany. Zmodyfikowany kod protokołu używa klucza publicznego Y, gdzie:

$$Y=g^x \bmod p$$

zaś x jest tajnym kluczem atakującego. Pomysł Adama Younga i Moti Yunga, wynalazców kleptografii, był (po pewnych uproszczeniach) następujący:

- W pierwszym wykonaniu protokołu, maszyna Alicji wybiera współczynnik a losowo, zapamiętuje go jednak do drugiego wykonania protokołu.
- W trakcie drugiego połączenia, nawiązywanego przy użyciu protokołu Diffiego-Helmana, liczba a nie jest wybierana losowo, ale obliczana jako

$$a'=\text{SHA-1}\ (Y^a \bmod p)$$

Zauważmy, jak w takim przypadku postępuje osoba posiadająca tajny klucz kleptograficzny x. Oblicza ona

$$\text{SHA-1}\ (A^x \bmod p)$$

Ponieważ $Y^a = g^{xa} \bmod p$ oraz $A^x = g^{ax} \bmod p$ atakujący otrzymuje tajny wykładnik a'. W kolejnym kroku oblicza uzgadniany klucz sesyjny, w dokładnie taki sam sposób, jak maszyna Alicji.
Przedstawione szczegóły techniczne pokazują, jak łatwy do zaimplementowania jest atak. Wystarcza w nim podsłuchanie protokołu uzgadniania kluczy aby możliwe było odszyfrowanie całej komunikacji prowadzonej następnie z użyciem klucza sesyjnego. W szczególności możliwe jest przejęcie połączenia w klasyczny sposób.

Problemy z SSL – kanały podprogowe

Piętą achillesową protokołu SSL, jest wykorzystanie losowych parametrów. Użycie takich parametrów wydaje się absolutnie niezbędne – inaczej atakujący, mógłby zgadnąć stan maszyny Alicji i w pewnym momencie na przykład przejąć połączenie. Z drugiej strony, już w 2003 roku Goh, Boneh, Pinkas i Golle z Uniwersytetu Stanford oraz

HP Labs, wskazywali na możliwość zaimplementowania tzw. kanału podprogowego w protokole SSL. Wykorzystywali w tym celu losowe *cookies*, przesyłane zgodnie z protokołem. Informacje na ten temat zostały przedstawione na konferencji ISC'2003 w Wielkiej Brytanii.
Rozwiązanie to korzystało z tajnych kluczy zawartych w kodzie zmodyfikowanej przeglądarki. Zagrożenia wynikające z ataku, były więc ograniczone – w razie wbudowania kodu w skompilowaną przeglądarkę i odkrycia tego faktu, wszyscy mieliby jednakowe możliwości przeprowadzenia ataku. W pewnym sensie mielibyśmy do czynienia z zachowaniem warunków „uczciwej konkurencji".

Postępy kleptografii

W 2005 roku, na konferencji ISC'2005 w Singapurze, Adam Young przedstawił możliwości związane z wykorzystanie Microsoft API, w celach kleptograficznych. Kolejny cios związany był z odkryciem ataków kleptograficznych na protokoły i urządzenia związane z e-wyborami. Metody skuteczne w stosunku do prawie wszystkich pojawiających się na rynku produktów, zostanią przez nas zaprezentowane w czerwcu br., na konferencji ETRICS'2006 we Freiburgu. Możliwe są rozszerzenia owych ataków i zastosowanie ich przeciwko mixom, serwerom używanym dla zapewnienia anonimizacji komunikacji.

Atak na SSL

Możliwe są dwie opcje uzgadniania wspólnego klucza w ramach protokołu SSL. Jedna metoda, to wykorzystanie protokołu Diffiego-Hellmana. Druga, to wybór sekretu przez *klienta* i przekazanie go *serwerowi* w postaci zaszyfrowanej algorytmem RSA. Zastosowanie protokołu Diffiego-Hellmana, implikuje możliwość wbudowania wspomnianego powyżej mechanizmu. Zastosowanie opcji z wykorzystaniem szyfrowania algorytmem RSA, zdaje się wykluczać tę możliwość. Algorytm RSA jest bowiem deterministyczny – nie ma w nim miejsca na losowy wybór, a tym samym stworzenie kanału, którym przekazywany jest sekret w sposób kleptograficzny.
Na pomoc atakującemu przychodzi jednak losowy ciąg Client.Random, wysyłany tekstem jawnym na początku protokołu przez klienta. Wykorzystując ten ciąg w sposób kleptograficzny można ustalić wartość sekretu – tak aby łamanie RSA nie było w ogóle potrzebne atakującemu.

Jak się obronić?

Na szczęście uodpornienie protokołu SSL przeciw kleptografii nie jest trudne. Co więcej, można zachować kompatybilność z dotychczasowym protokołem SSL/TLS, zdefiniowanym przez dokumenty RFC.
Pomysł, w istocie pochodzący od Davida Chauma – człowieka legendy współczesnej kryptografii, polega na zastosowaniu deterministycznej losowości (technika ta została przez niego użyta w protokole *Visual Voting*, jednak została „ukryta" w specyfikacji protokołu. Parametry losowe potrzebne są w protokołach aby zapewnić niemożność ich zgadnięcia przez osobę postronną i uniemożliwić tzw. *replay attack*. Cel ten może zostać osiągnięty w inny sposób: nieprzewidywalnym ciągiem bitów może być podpis cyfrowy danego uczestnika protokołu. O ile dodatkowo schemat podpisu jest deterministyczny (jak w przypadku podpisów RSA), w ciągu takim nie można ukryć żadnej wiadomości.
Zastosowanie podpisu cyfrowego jako generatora nieprzewidywalnych ciągów, ma jeszcze tę zaletę, że możliwe jest sprawdzenie, czy użyty parametr pseudolosowy, został utworzony w taki sposób.

Specyfikacja „łatki" na SSL

Aby zabezpieczyć się przed atakami kleptograficznymi należy:

1. uniemożliwić wymianę klucza, przy pomocy protokołu Diffie-Hellmana (można to zrobić po stronie serwera SSL)
2. zmodyfikować sposób generowania wartości *Client.Random* w ten sposób, aby była ona zależna (w sposób deterministyczny) od wartości *preMasterSecret* (ustalany wspólny sekret). Np., aby *Client.Random* = *hash* ('abc'+*preMasterSecret*), a następnie skompilować bibliotekę.

Co najważniejsze, serwer może wymusić na klientach stosowanie zabezpieczonej przeciwko atakom kleptograficznym przeglądarki. W celu kontroli, wystarczy wprowadzić drobną zmianę w oprogramowaniu serwera, która będzie polegała na sprawdzeniu, czy po otrzymaniu *preMasterSecret* zachodzi równość:

Client.Random = *hash* ('abc'+*preMasterSecret*)
Jeżeli ta równość nie będzie zachodzić, to serwer może zerwać połączenie.
Wydaje się, że łatka tego typu powinna być zastosowana w wypadku korzystania z bankowości internetowej i innych usług, w których po stronie klienta nie można oczekiwać profesjonalnej ochrony własnego komputera. Nic nie przemawia też przeciwko wbudowaniu odpowiednich zmian, do przyszłych wersji standardu.