УДК 681.3.07

РОЗШИРЕННЯ МОДЕЛІ ТИПІЗОВАНОЇ СПЕЦИФІКАЦІЇ ДЛЯ ФОРМАЛЬНОГО АНАЛІЗУ КРИПТОГРАФІЧНИХ ПРОТОКОЛІВ

П.О. КРАВЧЕНКО

У статті розглянуті основи формального аналізу протоколів, властивості безпеки протоколів та модель типізованої специфікації для формального аналізу. Оскільки складність криптографічних систем росте та протоколи вже не можна розглядати уособлено, без урахування контексту, у якому вони виконуються, ми розширюємо модель типізованої специфікації таким чином, що стає можливим проводити аналіз суттєво інших протоколів. Як приклад, буде розглянуто, як за допомогою розширеної моделі вдається знайти атаки на протокол Нідхема-Шредера з симетричним ключем.

Ключові слова: протокол, формальний аналіз, модель загроз, властивості безпеки, типізована специфікація

ВСТУП

Протокол (можливо вказівка на нормативний документ) — це набір правил і угод, які визначають комунікації між двома і більше учасниками. Ці учасники можуть бути кінцевими користувачами, процесами або обчислювальними системами. У криптографічних протоколах як мінімум одне повідомлення зашифровано. Криптографічні протоколи використовуються для встановлення безпечного з'єднання по небезпечним каналах зв'язку. Ці протоколи використовують криптографічні механізми для досягнення таких цілей, як конфіденційність, аутентифікація, цілісність, захист від повтору та інших. Відкриті мережі та розподілені системи уразливі для порушників, які можуть спотворити цілі протоколу.

Під атакою на протокол розуміється спроба проведення аналізу повідомлень протоколу і / або виконання не передбачених протоколом дій з метою порушення роботи протоколу та / або отримання інформації, що становить секрет його учасників.

Атака вважається успішною, якщо порушено хоча б одне із заявлених властивостей, які характеризують безпеку протоколу. В основі атак можуть лежати різні методи аналізу протоколів.

1.1. МОДЕЛЬ ВЛАСТИВОСТЕЙ, СЕРЕДОВИЩА ТА ПОРУШНИКА

Для визначення коректності захищеного протоколу ми спочатку повинні визначити модель, у якій ми будемо аналізувати протокол. Ми будемо спиратися на модель загроз Долева-Яо [1] – вона складається з трьох субмоделей: моделі властивостей, моделі порушника, моделі середовища.

Модель властивостей дозволяє зробити формалізацію цілей протоколу, які він гарантує досягти. Цілі безпеки також відомі як вимоги до протоколу або властивості безпеки.

Модель порушника описує учасника, який не обов'язково виконує правила протоколу. Його головною задачею є злом протоколу шляхом викривлення цілей протоколу (визначених моделлю властивостей). В моделі порушника деталізуються його можливості, дії, які він може робити для досягнення своєї мети. Модель порушника також називають моделлю загроз.

Модель середовища є описом всього навколишнього світу порушника (визначеного в моделі порушника). Модель середовища включає чесних учасників, що виконують правила протоколу. Модель середовища описує механізми комунікації між учасниками. Також ця модель повинна описувати будь-які інтереси суб'єктів реального світу, які мають вплив на поведінку (чи гарантії безпеки) протокола. Наприклад, це може бути моделювання внутрішніх властивостей мережі, таких як шум або правила маршрутизації.

1.2. ВЛАСТИВОСТІ, ЩО ХАРАКТЕРИЗУЮТЬ БЕЗПЕКУ ПРОТОКОЛІВ

Найважливішою властивістю криптографічної системи є забезпечення різних функції безпеки, для реалізації яких застосовуються криптографічні протоколи. Властивості протоколів, що характеризують їх стійкість до різних атак, формулюють як цілі (goals) або вимоги до протоколів. Перелік, визначення та тлумачення цілей безпеки дається в документах міжнародної організації IETF [7].

Під властивостями безпеки в документах IETF в даний час розуміються наступні 20 цілей, згруповані в 10 груп (табл. 1). Наведемо визначення деяких з перерахованих там властивостей.

1.3. ФОРМАЛЬНІ МОДЕЛІ КРИПТОПРОТОКОЛІВ

Переважна більшість використовуваних на сьогоднішній день формальних моделей криптопротоколів так чи інакше спираються на модель порушника, запропоновану Долевим та Яо [1]. Разом з даною моделлю автори запропонували метод формального аналізу протоколів. Незважаючи на те, що безпосередньо цей метод аналізу протоколів виявив в собі занадто жорсткі обмеження, створена модель, в якій порушник повністю володіє каналом передачі даних (будучи здатним до читання, перехопленню та отриманих

Таблиця 1

Властивості безпеки протоколів

№	Код	Властивість безпеки
1	G1 G2 G3	Автентифікація суб'єкта Автентифікація повідомлення Захист від повтору
2	G4 G5	Неявна (прихована) автентифікація одержувача Автентифікація джерела
3	G6	Авторизація (довіреної третьою стороною)
4	G7 G8 G9 G10 G11	Автентифікація ключа Підтвердження правильності ключа Захищеність від читання тому Формування нових ключів Захищена можливість домовитися про параметри безпеки
5	G12	Конфіденційність
6	G13 G14	Забезпечення анонімності при прослуховуванні Забезпечення анонімності при роботі з іншими учасниками
7	G15	Обмежена захищеність від атак типу відмова в обслуговуванні
8	G16	Незмінність відправника
9	G17 G18 G19	Підзвітність Доказ відправки Доказ отримання
10	G20	Безпечна часова властивість

повідомлень) точно сформулювала набір можливостей реального порушника [MD].

Основні положення цієї моделі зводяться до наступного.

1) Противник може використовувати будь-які доступні йому комбінації стандартних операцій для побудови нових повідомлень з тих, які йому відомі. Противник завжди знає структуру усіх повідомлень, що були передані.

2) Всі криптографічні алгоритми володіють властивістю довершеності, тобто противник не взмозі отримати доступ до повідомлення без відповідного таємного ключа.

3) Противник володіє повним контролем над усіма каналами зв'язку. Він може прослуховувати всі повідомлення, видаляти повідомлення з каналів зв'язку і перенаправляти їх іншим адресатам, формувати і відправляти будь-яким учасникам повідомлення.

1.4. КЛАСИФІКАЦІЯ ФОРМАЛЬНИХ МЕТОДІВ АНАЛІЗУ

В даний час існують десятки формальних методів аналізу криптопротоколів. В [8] наводиться їх класифікація. Відповідно до неї, основу формального аналізу протоколів можуть становити:

- Моделювання та перевірка роботи протоколу. Для цієї мети корисно використовувати спеціалізовані мови і інструментарії, які не створювалися для аналізу криптопротоколів;
- Створення експертних систем, які розробники криптопротоколів можуть застосувати для апробування різних сценаріїв функціонування криптопротоколів;
- Моделювання вимог до сімейства криптопротоколів. При цьому можна вжити логіку, розроблену спеціально для аналізу таких властивостей криптопротоколів, як «знання» і «довіра»;
- Розробка формальних моделей, заснованих на алгебраїчних властивості криптографічних систем.

Кожен з перерахованих підходів не прив'язаний до лежачих в основі криптопротоколів механізмам, а спрямований тільки на аналіз логіки роботи протоколу.

В [9] була запропонована інша класифікація. У ній методи аналізу розбиваються всього на дві групи:

- Методи побудови логічних висновків (Inference-construction methods), які використовують логіки, засновані на поняттях знання і довіри.
- Методи конструювання можливих атак проникнення в захищену систему (Attack-construction methods) з використанням алгебраїчних властивостей алгоритмів, що лежать в основі протоколів.

2. ТИПІЗОВАНІ СПЕЦИФІКАЦІЇ БЕЗПЕКИ ПРОТОКОЛУ

Однією з різновидів формального аналізу є типізовані специфікації безпеки протоколів. Нижче ми наведемо типізовану модель безпеки протоколів, яка була запропонована у [2]. Потім ми покажемо, як можна розширити цю модель, для досягнення більшої гнучкості.

2.1. БАЗОВІ ПОНЯТТЯ

Ми будемо розглядати криптосистему як набор агентів з'єднання. Безпеку протоколу описує поведінка цих агентів, які звуться «*ролями*». Таким чином специфікація безпеки протоколу безпеки зводиться до аналізу поведінку ролей, які беруть участь у протоколі. Навести визначення РОЛЕЙ (Черемушкин).

Базові набори. Набори, які формують основу нашої конструкції та сконструйовані з наступних елементів: R (позначає набір ролей, наприклад, {A, B, C}, де A, B, C є іменами ролей), N (позначає набір міток часу, наприклад {Na, Nb, Nc, Nt}, де Na, Nb, Nc є мітками часу, які були згенеровані A, B, C відповідно, а позначення Nt використовується як мітка реального часу в системі), та F (позначає набір імен функцій, наприклад *sym*, позначає симетричне шифрування, а *asym* – асиметричне шифрування).

Криптографічні примітиви. Згідно моделі Долева-Яо, усі криптографічні примітиви вважаються безумовно стійкими.

Комунікаційна модель. Комунікаційна модель також відповідає моделі безпеки Долева-Яо, де будь-яка роль може читати повідомлення, що розповсюджуються по каналу зв'язку, та будь-яка роль може відіслати повідомлення по каналу.

2.2. СПЕЦИФІКАЦІЯ БЕЗПЕЧНОСТІ ПРОТОКОЛУ

Ключі шифрування в повідомленні криптографічного протоколу мають наступний вигляд:

Keys: K:: = k (ключ сесії)
| sh A (*розподілений* ключ)
| pk A (*відкритий* ключ)
| sk A (*секретний* ключ)

Символ K використовується для позначення всього діапазону ключів, що з'являється в повідомленнях.

Повідомлення, або *Message Term*, позначене як M, визначається таким чином:

$$Message\ Term: M ::= . \mid R \mid N \mid K \mid F(M) \mid \{M\}_m .$$

Для позначення порожнього термального повідомлення, використовується символ ".". Тип алгоритму шифрування (симетричного або асиметричного) та ключ, який використовувався, позначаються індексом поміщеним після дужок "{}". Якщо індекс не є функцією F, тоді шифрування вважається симетричним. Крім того, якщо контекст дозволяє (не призводить до плутанини) опускається специфікація функції, та залишається тільки сеансовий ключ шифрування.

Треба розрізняти відправлене та отримане повідомлення, тому визначається два предиката *send*, та *recv*: $R \times R \times M$ для позначення відправки та отримання повідомлень, що мають ролі джерела та ролі цілі. Композиція та декомпозиція повідомлень визначені індуктивно за наступними правилами:

$$send(r,r',t_1) \wedge send(r,r',t_2) \Leftrightarrow send(r,r',(t_1,t_2)),$$
$$send(r,r',t) \wedge send(r,r',f(t_1,\ldots,t_n)) \Leftrightarrow$$
$$\Leftrightarrow send(r,r',(t,f(t_1,\ldots,t_n))),$$
$$recv(r,r',t_1) \wedge recv(r,r',t_2) \Leftrightarrow recv(r,r',(t_1,t_2)),$$
$$\Leftrightarrow recv(r,r',(t,f(t_1,\ldots,t_n))),$$

де $r,r' \in R, t,t_1,\ldots,t_n \in M$ та $f \in F$. r та r' використовуються для позначення ролі джерела та ролі цілі, відповідно.

Таким чином, *роль специфікації* визначається у вигляді наборів предикатів *send* та *recv*, використовуючи індекс $i \in I$ для розрізнення подібних випадків:

$$RoleSpec =$$
$$= \{send_i(r,r',t), recv_i(r,r',t) \mid t \in M, i \in I, r,r' \in R\}.$$

Визначивши роль специфікації, можливо визначити специфікації протоколу, що описують поведінку декількох ролей, як функцію $ProtSpec = R \rightarrow RoleSpec$.

2.3. ТИПОВА СПЕЦИФІКАЦІЯ БЕЗПЕКИ ПРОТОКОЛУ

Дана типова специфікація безпеки протоколу має в своїй основі базові типи – *Basic Types*, що визначені наступним чином:

$$Basic\ Types: \tau ::= r(role\ type)$$
$$\mid n\ (nonce\ type)$$
$$\mid k\ (session\ key\ type)$$
$$\mid sh\ AB(shared\ key\ type)$$
$$\mid pk\ A(public\ key\ type)$$
$$\mid sk\ A(secret\ key\ type)$$

Порівнюючи визначення *Повідомлення* з розділу 2.2 з попереднім визначенням, бачимо, що літери без курсиву означають типи відповідних компонентів *повідомлення*. Символ τ використовується для позначення всіх можливих базових типів.

Typed Message, або *Typed Message Term*, позначені як tM, побудовані з використанням базових типів τ і мають наступне визначення (символ "." використовується для позначення порожнього термального повідомлення

$$Typed\ Message\ Term: tM ::=$$
$$. \mid \tau \mid F(tM) \mid (tM,tM) \mid \{tM\}_{tM}$$

Визначення *типової специфікації ролі* використовує предикати *tsend*, *trecv* : $R \times R \times tM$ для позначення відправлених та отриманих *Typed Messages* від джерела ролі до призначення ролі, та має визначення специфікації ролі схоже з попереднім розділом:

$$TRoleSpec =$$
$$= \{tsend_i(r,r',t), trecv_i(r,r',t) \mid t \in tM, i \in I, r,r' \in R\}$$

Композиція та декомпозиція правил типізованих повідомлень визначаються, як:

$$tsend(r,r',t_1) \wedge tsend(r,r',t_2) \Leftrightarrow tsend(r,r',(t_1,t_2)),$$
$$tsend(r,r',t) \wedge tsend(r,r',f(t_1,\ldots,t_n)) \Leftrightarrow$$
$$\Leftrightarrow tsend(r,r',(t,f(t_1,\ldots,t_n))),$$
$$trecv(r,r',t_1) \wedge trecv(r,r',t_2) \Leftrightarrow trecv(r,r',(t_1,t_2)),$$
$$trecv(r,r',t) \wedge trecv(r,r',f(t_1,\ldots,t_n)) \Leftrightarrow$$
$$\Leftrightarrow trecv(r,r',(t,f(t_1,\ldots,t_n))),$$
де $r,r' \in R, t,t_1,\ldots,t_n \in tM,\ f \in F.$

Аналогічно з *TRoleSpec*, визначимо типову специфікацію безпеки протоколу як функцію $TProtSpec = R \rightarrow TRoleSpec$.

2.4. ПЕРЕТВОРЕННЯ

Для моделі існуючих протоколів у рамках типової структури, нам потрібна функція, яка перетворює *не типізовані* повідомлення (тобто M) на типізовані (тобто tM).

Перетворення термального повідомлення $t,t_1,\ldots,t_n \in M$, та $f \in F$, в типове термальне повідомлення визначено як:

$$MTr(t)=\begin{cases} r, & if\ t\equiv r\in R \\ n, & if\ t\equiv n\in N \\ k, & if\ t\equiv k\in M \\ sh\ A\ B, & if\ t\equiv sh\ A\ B\in M \\ pk\ A, & if\ t\equiv pk\ A\in M \\ sk\ A, & if\ t\equiv sk\ A\in M \\ f(MTr(t_1),...,MTr(t_n)), & if\ t\equiv f(t_1,...,t_n) \\ (MTr(t_1),.MTr(t_2)), & if\ t\equiv(t_1,.t_2) \\ \{MTr(t_1)\}_{MTr(t_2)}, & if\ t\equiv\{t_1\}_{t_2} \end{cases}$$

Для перетворення *специфікації ролі* в *типову специфікацію ролі* використовується *функція трансформації ролі*: $RTr = TRoleSpec \to RoleSpec$.

У випадку трансформації повідомлення необхідно визначити функцію трансформації ролі для $p, p_1, p_2 \in RoleSpec, t \in M, i \in I, r, r' \in R$ як:

$$RTr(\rho)=\begin{cases} (RTr(\rho_1),RTr(\rho_2)), & if\ \rho\equiv(\rho_1,\rho_2) \\ tsend_i(r,r',MTr(t)), & if\ \rho\equiv send_i(r,r',t). \\ trecv_i(r,r',MTr(t)), & if\ \rho\equiv recv_i(r,r',t) \end{cases}$$

3. АНАЛІЗ ТИПОВОЇ БЕЗПЕКИ ПРОТОКОЛУ

Для того, щоб проаналізувати захищеність протоколів від *структурних атак*, потрібно спочатку формалізувати обґрунтовані атаки (*повтору* і *типової плутанини*) з використанням типових структур.

3.1. ФОРМАЛІЗАЦІЯ АТАКИ

У цьому розділі буде показано як формалізуються *структурні атаки* за допомогою використання типових специфікацій безпеки протоколів.

Для опису атаки повтору ролі $r \in R$ в специфікації протоколу, спочатку будується набір всіх «*відправлених*» повідомлень будь-якого джерела ролі $r' \in R$ та призначення ролі $r'' \in R - \{r\}$:

$$allsentMsgExluding(r) = \bigcup_{r'\in R, r''\in R-\{r\}} \{t \mid send_i(r',r'',t) \in T\ \mathrm{Pr}otSpec(r')\}$$

Аналогічно визначається набор всіх «*отриманих*» повідомлень, що наведені в специфікації протоколу, з тими ж $r \in R$ та $r' \in R$:

$$recvdMsg(r) = \bigcup_{r'\in R} \{t \mid recv_i(r,r',t) \in T\ \mathrm{Pr}otSpec(r)\}$$

Для даного набору всіх повідомлень в специфікації протоколу не призначено *r*, та набір повідомлень, наданий в специфікації *r*, який може бути отриманий через *r*, для визначення ролі *r*, відкритий для атак *повтору*. Треба тільки знайти повідомлення чи суб-повідомлення в *recvdMsg*, який є еквівалентним повідомленню чи суб-повідомленню в *allsentMsgExcluding*. Формально, використовується предикат *REPLAY*, для позначення факту того, що роль $r \in R$ відкрита для атак *повтору*:

$$REPLAY(r) \Leftrightarrow \exists_{t=t'} t_r \in recvdMsg(r),$$
$$t_s \in allsentMsgExcluding(r), t \in P(t_r), t' \in P(t_s).$$

Так як атаки типу *плутанини* є результатом прийняття зашифрованих повідомлень, які містять ключі (наприклад, ключі сесії, згенеровані сервером третьою стороною), ці атаки моделюються аналогічно до атак типу *повтор*.

Для визначення, чи є роль $r \in R$ відкритою для атак типу *плутанини*, спочатку будується набір всіх специфікацій «*відправлених*» повідомлень для всіх ролей, що беруть участь у протоколі. Повідомлення, призначені для ролі *r,* не видаляються (як це робилося у випадку моделювання атак повтору), оскільки ці повідомлення можуть бути повторно використані в різних умовах, створюючи, таким чином, можливість для атак типу *мішанини*:

$$allsentMsg = \bigcup_{r'.r''\in R} \{t \mid send_i(r',r'',t) \in T\ \mathrm{Pr}otSpec(r')\}.$$

Таким чином, ми предикат *BASIC-TYPEFLAW* використовується для позначення того, що роль $r \in R$ є відкритою для атак типу плутанини:

$$BASIC\_TYPEFLAW(r) \Leftrightarrow$$
$$\exists t_r \in recvdMsg(r), t_s \in allsentMsg, t \in P(t_r),$$
$$t' \in P(t_s) \wedge \exists t_1, t_2, t_3 \subset t, t_1', t_2' \subset t', f \in t, bt \in \tau - \{k\},$$
$$t = \{t_1, k, t_2,\}_{f(t_3)} \wedge t' = \{t_1', bt, t_2',\}_{f(t_3)} \wedge |t_1| =$$
$$= |t_1'| \wedge |t_2| = |t_2'|,$$

де оператор $\subset$ використовується для позначення того, що термальне повідомлення є субтермом іншого повідомлення, оператор $|t|$ визначає довжину термального повідомлення *t*. Функція шифрування *f* і ключ шифрування t_3 є однаковими для відправлених та отриманих повідомлень.

3.2. АНАЛІЗ БЕЗПЕКИ ПРОТОКОЛУ НЬЮМАНА-СТАББЛЕБАЙН

Нижче буде показано як, використовуючи типову специфікацію протоколу, можна знайти атаки на відомий протокол Ньюмана-Стабблебайн [3], що описані у [4]. Надамо регулярну специфікацію протоколу:

$$NS(A)=\left\{\begin{array}{l} send_1(A,B,(A,Na), \\ recv_2(S,A,(\{B,Na,k,Nt\}_{shAS},\{A,k,Nt\}_{shBS},Nb)), \\ send_3(A,B,(\{A,k,Nt\}_{shBS},\{Nb\}_k)) \end{array}\right\},$$

$$NS(B)=\left\{\begin{array}{l} recv_1(A,B,(A,Na), \\ send_2(B,S,(B\{A,Na,Nt\}_{shBS},Nb)), \\ recv_3(A,B,(\{A,k,Nt\}_{shBS},\{Nb\}_k)) \end{array}\right\},$$

$$NS(S)=\left\{\begin{array}{l} recv_1(B,S,(B\{A,Na,Nt\}_{shBS},Nb)), \\ send_2(S,A,(\{B,Na,k,Nt\}_{shAS},\{A,k,Nt\}_{shBS},Nb)) \end{array}\right\}.$$

Застосовуючи функцію трансформації ролі (15), маємо наступну типову специфікацію:

$$RTr(NS(A))=\begin{Bmatrix} tsend_1(A,B,(r,n),\\ recv_2(S,A,(\{r,n,k,n\}_{shAS},\{r,k,n\}_{shBS},n)),\\ send_3(A,B,(\{r,k,n\}_{shBS},\{n\}_k)) \end{Bmatrix}$$

$$RTr(NS(B))=\begin{Bmatrix} trecv_1(A,B,(r,n),\\ tsend_2(B,S,(r,\{r,n,n\}_{shBS},n)),\\ trecv_3(A,B,(\{r,k,n\}_{shBS},\{n\}_k)) \end{Bmatrix}$$

$$RTr(NS(S))=\begin{Bmatrix} trecv_1(B,S,(r,\{r,n,n\}_{shBS},n)),\\ tsend_2(S,A,(\{r,n,k,n\}_{shAS},\{r,k,n\}_{shBS},n)) \end{Bmatrix}.$$

Якщо проналізувати типову специфікацію для ролі B, предикат $REPLAY(B)$ буде стійким, наприклад, для $trecv_1(A, B, (r, n))$, де (r, n) повідомлення можуть бути витягнуті з $tsend_2(B, S, (r, \{r, n, n\}_{shBS}, n))$ повідомлення. Це веде до простої атаки *повтору*, яка створюється шляхом відправки до B повідомлення, яке він сам згенерував.

Через схожість структури повідомлень, предикат $BASIC\text{-}TYPEFLAW(B)$ також буде стійким в багатьох випадках. Наприклад, в повідомленні $trecv_3(A, B, (\{r, k, n\}_{shBS}, \{n\}_k))$, терм $\{r, k, n\}_{shBS}$ може бути згенерований у вигляді $tsend_2(B, S, (r, \{r, n, n\}_{shBS}, n))$ таким чином, перша мітка часу відсилається до $\{r, n, n\}_{shBS}$, терм повідомлення стає ключем.

4. РОЗШИРЕННЯ МОДЕЛІ АНАЛІЗУ

Одним з недоліків даної моделі є те, що вона розглядує без контексту, в якому працює криптосистема. Ми пропонуємо використовувати розширену модель аналізу, щоб врахувати контекст запуску протоколу. Розширення моделі проводиться шляхом наведення та аналізу типових специфікацій попередніх та наступних запусків протоколу. Щоб довести, що розширення моделі призводить до якісно нових результатів, проаналізуємо протокол Нідхема-Шредера з симетричним ключем [5], побудувавши його типову специфікацію.

Наведемо регулярну специфікацію протоколу Нідхема-Шредера із симетричним ключем:

$$NS(A)=\begin{Bmatrix} send_1(A,S,(A,B,Na)),\\ recv_2(S,A,(\{N_A,K,B,\{K,A\}_{K_{BT}}\}_{K_{AT}})),\\ send_3(A,B,(\{K,A\}_{K_{BT}})),\\ recv_4(B,A,(\{M,N_B\}_K)),\\ send_5(A,B,(\{M',N_b-1\}_K)) \end{Bmatrix}$$

$$NS(B)=\begin{Bmatrix} recv_1(A,B,(\{K,A\}_{K_{BT}})),\\ send_2(B,A,(\{M,N_B\}_K)),\\ recv_3(A,B,(\{M',N_b-1\}_K)) \end{Bmatrix}$$

$$NS(S)=\begin{Bmatrix} recv_1(A,S,(A,B,Na)),\\ send_2(S,A,(\{N_A,K,B,\{K,A\}_{K_{BT}}\}_{K_{AT}}) \end{Bmatrix}.$$

Застосовуючи функцію трансформації, маємо наступну типову специфікацію:

$$RTr(NS(A))=\begin{Bmatrix} tsend_1(A,S,(r,\ r,\ n),\\ recv_2(S,A,(\{n,k,r,\{k,r\}_{shBT}\}_{shAT})),\\ send_3(A,B,(\{k,r\}_{shBT}\ )),\\ recv_4(B,A,(\{m,n\}_{shAB}\ )),\\ send_5(A,B,(\{m,n\}_{shAB}\ )) \end{Bmatrix}$$

$$RTr(NS(B))=\begin{Bmatrix} trecv_1(A,B,(\{k,r\}_{shBT}),\\ tsend_2(B,A,(\{m,n\}_{shAB}\ )),\\ trecv_3(A,B,(\{m,n\}_{shAB}\ )) \end{Bmatrix}$$

$$RTr(NS(S))=\begin{Bmatrix} trecv_1(A,S,(r,\ r,\ n),\\ tsend_2(S,A,(\{n,k,r,\{k,r\}_{shBT}\}_{shAT})) \end{Bmatrix},$$

де $shAB = k$.

Аналізуючи типову специфікацію для ролі B у стандартній моделі аналізу, ми не можемо в явному вигляді знайти умови, в яких предикати $REPLAY(B)$ та $BASIC\text{-}TYPEFLAW(B)$ буде стійкими.

Тепер ми будемо застосовувати розширену модель аналізу і розглянемо протокол у контексті. Наведемо типову специфікацію будь-якого попереднього сеансу між E (зловмисником) та B:

$$RTr(NS(A))=\begin{Bmatrix} tsend_1(E,S,(r,\ r,\ n),\\ recv_2(S,E,(\{n,k,r,\{k,r\}_{shBT}\}_{shET})),\\ send_3(E,B,(\{k,r\}_{shBT}\ )),\\ recv_4(B,E,(\{m,n\}_{shEB}\ )),\\ send_5(E,B,(\{m,n\}_{shEB}\ )) \end{Bmatrix},$$

$$RTr(NS(B))=\begin{Bmatrix} trecv_1(E,B,(\{k,r\}_{shBT}),\\ tsend_2(B,E,(\{m,n\}_{shEB}\ )),\\ trecv_3(E,B,(\{m,n\}_{shEB}\ )) \end{Bmatrix},$$

де $shEB = k$.

Якщо розглянути цей сеанс зв'язку між E та B (можливо також A та B – згідно моделі Долева-Яо, зловмисник має доступ до усіх ключевих даних, що були використані раніше та вже вважаються застарілими) та A і B, ми визначимо, що наприклад, для $trecv_1(A,B,(\{k,r\}_{shBT})$ блок $\{k,r\}_{shBT}$ може бути витягнутий з $send_3(E,B,(\{k,r\}_{shBT}\ ))$ попереднього сеансу протоколу. Це веде до простої атаки *повтору*, у якої Боб отримує старий ключ. Ця атака дуже критична, бо вона руйнує усю криптосистему – зловмиснику достатньо встановити сеанс зв'язку з B у будь-який час до його сеансу з A. Таким чином, за допомогою розширеної моделі ми зможемо легко знайти атаку Денінга-Сакко [6] на цей протокол.

ВИСНОВКИ

Ми розглянули основи формального аналізу криптографічних протоколів, модель безпеки Долева-Яо, класифікацію методів аналізу та

цілей безпеки. Був детально розглянутий метод формального аналізу, що є різновидом техніки статичного аналізу. Серед сильних сторін цього методу є простота побудови, та можливість автоматичного аналізу. Одним з його недоліків є те, що він не бере до уваги контекст, у якому пряцює криптографічна система. Цей метод був розширений таким чином, що стало можливим знайти атаку на протокол Нідхема-Шредера з симетричним ключем (що неможливо з використанням звичайної техніки аналізу).

Література

[1] *Dolev, D., Yao, A.,* On the security of public key protocols, IEEE Transactions on Information Theory, IT-29, 1983, pp. 198-208.

[2] *Bela G. Ignat I.,* An Abstract Model for Security Protocol Analysis, WSEAS Transactions on Computers, Vol. 6 (2007) , p. 207—215.

[3] *B. Clifford Neumann and Stuart G. Stubblebine.* A note on the use of timestamps as nonces. Operating Systems Review, 27(2):10—14, april 1993.

[4] *Tzonelih Hwang, Narn-Yoh Lee, Chuang-Ming Li, Ming-Yung Ko, and Yung-Hsiang Chen.* Two attacks on Neumann-Stubblebine authentication protocols. Information Processing Letters, 53:103 – 107, 1995.

[5] *R. Needham and M. Schroeder.* Using encryption for authentication in large networks of computers. Communications of the ACM, 21(12), December 1978.

[6] *Denning D., Sacco G.,* Timestamps in key distributed protocols. Communication of the ACM 24, 1981 (8): 533—535

[7] *E. Rescorla and B. Korver.* RFC 3552: Guidelines for Writing RFC Text on Security Considerations, July 2003.

[8] *C. Meadows,* Formal Methods for Cryptographic Protocol Analysis: Emerging Issues and Trends, IEEE Journal On Selected Areas In Communications, vol. 21, no. 1, 2003.

[9] *S. Gritzalis, D. Spinellis, P. Georgiadis,* Security Protocols over Open Networks and Distributed Systems: Formal Methods for their Analysis, Design, and Verification, Computer Communications, 22(8): 695—707, May 1999.

Надійшла до редколегії 18.05.2011

Кравченко Павло Олександрович, аспірант кафедри БІТ ХНУРЕ. Область наукових інтересів: асиметрична криптографія, ІВК на ідентифікаторах

УДК 681.3.07

Расширение модели типизированной спецификации для формального анализа криптографических протоколов / П.А. Кравченко // Прикладная радиоэлектроника: науч.-техн. журнал. – 2011. Том 10. № 2. – С. 189–197.

В статье рассмотрены основы формального анализа протоколов, свойства безопасности протоколов и модель типизированной спецификации для формального анализа. Предложено расширение типизированной спецификации, которое позволяет проводить анализ большего количества протоколов.

Ключевые слова: протокол, формальный анализ, модель угроз, свойства безопасности, типизированная спецификация

Табл. 01. Библиогр.: 09 назв.

UDC 681.3.07

Extending a model of typified specification for formal cryptographic protocol analysis / P.A. Kravchenko // Applied Radio Electronics: Sci. Journ. – 2011. Vol. 10. № 2. – P. 189–197.

The paper reviews the basis of formal analysis of protocols, security features and typed model specification for protocol formal analysis. Extension of the typified specification which allows analysis of bigger amount of protocols is proposed.

Keywords: protocol, formal analysis, model of threats, security goals, typified specification.

Tab. 01. Ref.: 09 items.