

Формальная верификация криптографических протоколов с использованием Proverif

Винарский Евгений

Институт системного программирования

28 июля 2021

Слабые места защиты информации

- Атаки на архитектуру (Криптографическая система не может быть надежнее использованных в ней отдельных алгоритмов шифрования)
- Атаки на конкретные реализации
 - Переполнение буферов
 - Не стёртая до конца секретная информация
- Атаки на сетевое оборудование
- Атаки на пользователей
- Атаки с использованием побочных каналов
- ...

Для того чтобы преодолеть систему защиты, достаточно взломать любой из ее компонентов

- Алгоритмы симметричного шифрования
- Алгоритмы асимметричной криптографии (выработка общих сессионных ключей и т.д.)
- Датчик псевдослучайных чисел
- Алгоритмы хэширования

В этом блоке считаем, что все криптографические примитивы не могут быть скомпрометированы раньше, чем перестанут использоваться

Какие достоинства и недостатки асимметричной криптографии?

- **Модель атаки** Возможности противника по взаимодействию с системой
- **Ресурсы противника** Предположения о вычислительных и информационных ресурсах противника
- **Угроза** Задача противника по нарушению свойств безопасности

Уязвимости криптосистемы возникают, если неправильно выбраны

- 1 модель атаки
- 2 угроза
- 3 предположения о ресурсах

Модель противника (2)

- *Пассивный противник* (противник может читать зашифрованные пересылаемые данные в открытом канале)
- *Dolev-Yao (Активный) Противник* может:
 - читать сообщения в канале
 - модифицировать сообщения в канале
 - удалять сообщения из канала
- Противник, учитывающий временные задержки (может определить, какая именно проверка не прошла, ...)
- ...

Рассуждать о стойкости криптосистемы можно только в терминах модели противника

Протокол (алгоритм) Диффи-Хеллмана

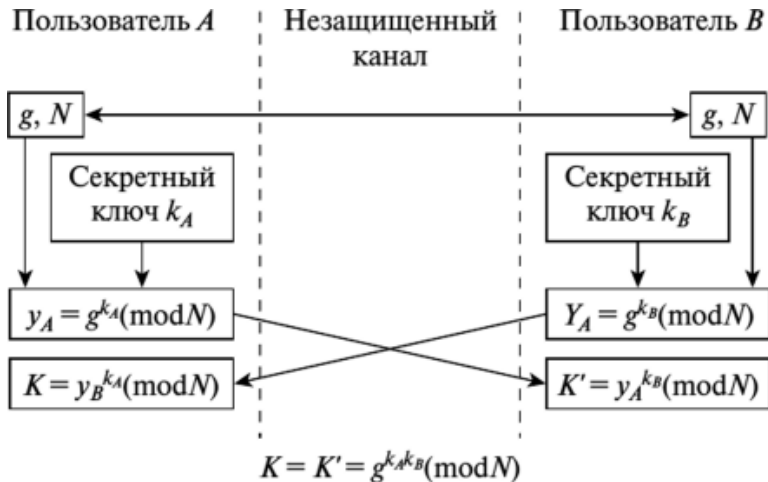


Схема протокола Диффи-Хеллмана

Модель противника

Протокол Диффи-Хеллмана при активном противнике

- E – пассивный противник, слушающий незащищённый канал
- E известно значение g , N и y_A, y_B

Угроза: Противник E узнал выработанный общий ключ
 $K = g^{k_A k_B} \pmod{N}$

Модель противника

Протокол Диффи-Хеллмана при активном противнике

- E – пассивный противник, слушающий незащищённый канал
- E известно значение g , N и y_A, y_B

Угроза: Противник E узнал выработанный общий ключ

$$K = g^{k_A k_B} \pmod{N}$$

Пусть E скомпрометировал $K = g^{k_A k_B} \pmod{N}$, тогда:

- узнал k_A
 - решил задачу дискретного логарифмирования, т.е. вычислил k_A из уравнения $y_A = g^{k_A} \pmod{N}$
- узнал k_B
 - решил задачу дискретного логарифмирования, т.е. вычислил k_B из уравнения $y_B = g^{k_B} \pmod{N}$
- узнал $k_A * k_B$

Протокол Диффи-Хэллмана стойкий по отношению к пассивному противнику

Свойства безопасности протоколов выработки общих ключей обмена

- Аутентификация
 - Ложная аутентификация
 - Unknown key share (Неизвестный общий ключ)
- Установление одинаковых ключей
- Секретность ключей обмена
- Уникальность установленных ключей обмена
- Forward secrecy (Прямая секретность)
- Backward secrecy (Обратная секретность)

Протокол Нидхема-Шрёдера

Протокол выработки общего сессионного ключа (N_A, N_B)

Alice ($pk_A, priv_A$)		Bob ($pk_B, priv_B$)
<ul style="list-style-type: none">- генерация N_A- шифрует $\langle A, N_A \rangle_{pk(B)}$	$\langle A, N_A \rangle_{pk(B)}$ ----->	
	$\langle N_A, N_B \rangle_{pk(A)}$ <-----	<ul style="list-style-type: none">- дешифровка сообщения $\langle A, N_A \rangle_{pk(B)}$- $\langle N_A, N_B \rangle_{pk(A)}$
<ul style="list-style-type: none">- дешифровка сообщения $\langle N_A, N_B \rangle_{pk(A)}$- $\langle N_B \rangle_{pk(B)}$	$\langle N_B \rangle_{pk(B)}$ ----->	
(N_A, N_B) -- общий секрет		

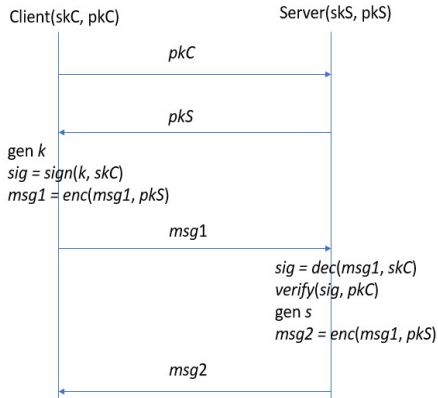
Уязвимость протокола Нидхема-Шрёдера

Alice	Intruder (pkI, privI)	Bob
<ul style="list-style-type: none">- генерация N_A- шифрует $\langle A, N_A \rangle_{pk(I)}$ <p>$\langle A, N_A \rangle_{pk(I)}$ -----></p>		
	<ul style="list-style-type: none">- дешифрует $\langle A, N_A \rangle_{pk(I)}$- шифрует $\langle A, N_A \rangle_{pk(B)}$ <p>$\langle A, N_A \rangle_{pk(B)}$ -----></p>	
		<ul style="list-style-type: none">- дешифровка сообщения $\langle A, N_A \rangle_{pk(B)}$- $\langle N_A, N_B \rangle_{pk(A)}$ <p>$\langle N_A, N_B \rangle_{pk(A)}$ <-----</p>

Уязвимость протокола Нидхема-Шрёдера (2)

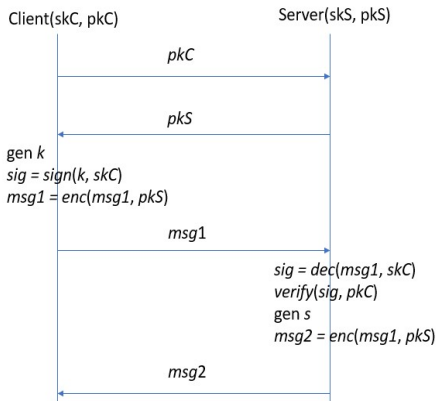
	$\langle NA, NB \rangle_{pk(A)}$ \leftarrow	
<ul style="list-style-type: none">- дешифровка сообщения $\langle NA, NB \rangle_{pk(A)}$- $\langle NB \rangle_{pk(I)}$ $\langle NB \rangle_{pk(I)} \rightarrow$		
	<ul style="list-style-type: none">- дешифровка $\langle NB \rangle_{pk(I)}$ $\langle NB \rangle_{pk(B)} \rightarrow$	
(NA, NB) -- общий секрет, который известен I		

Пример 1: код на Proverif



```
fun Exp(bitstring,bitstring,bitstring):bitstring.  
fun Sign(bitstring,bitstring):bitstring.  
  reduc forall msg:bitstring,sign_key:bitstring;  
    CheckSign(Sign(msg, sign_key), Exp(xCurve, xBase, sign_key)) = msg.  
fun Encrypt(bitstring,bitstring):bitstring.  
  reduc forall a0:bitstring,a1:bitstring;  
    Decrypt(Encrypt(a0,a1),a1) = a0.  
fun A_Encrypt(bitstring,bitstring):bitstring.  
  reduc forall a0:bitstring,a_key:bitstring;  
    A_Decrypt(A_Encrypt(a0,Exp(xCurve, xBase, a_key)),a_key) = a0.
```

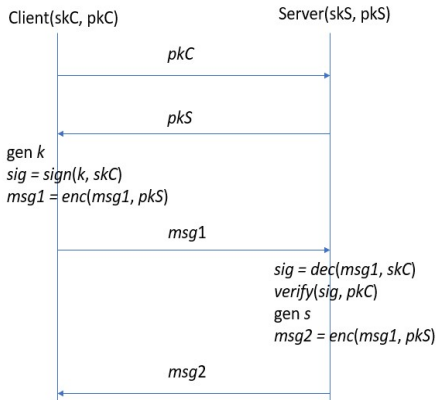
Пример 1: код на Proverif (2)



```
let Client(skC:bitstring, pkC:bitstring) =
  out(c, pkC);
  in(c, pkS:bitstring);
  (* msg1 *)
  new k:bitstring;
  let sigC = Sign(k, skC) in
  let msg1 = A_Encrypt(sigC, pkS) in
  out(c, msg1);
  (* msg2 *)
  in(c, msg2:bitstring);
  let s = Decrypt(msg2, k) in
  event Client_end(s).
```

```
let Server(skS:bitstring, pkS:bitstring) =
  in(c, pkC:bitstring);
  out(c, pkS);
  (* msg1 *)
  in(c, msg1:bitstring);
  let sigC = A_Decrypt(msg1, skS) in
  let k = CheckSign(sigC, pkC) in
  (* msg2 *)
  (*new s:bitstring;*)
  let msg2 = Encrypt(s, k) in
  out(c, msg2);
  event Server_end(s).
```

Пример 1: описание протокола дизъюнктами Хорна



Дизъюнкты Хорна, представляющие протокол

$\text{attacker}(pk(x))$
 $\Rightarrow \text{attacker}(\text{pencrypt}(\text{sign}(k[pk(x)], sk_A[]), pk(x)))$
 $\text{attacker}(\text{pencrypt}(\text{sign}(y, sk_A[]), pk(sk_B[])))$
 $\Rightarrow \text{attacker}(\text{sencrypt}(s, y))$

Дизъюнкты Хорна, представляющие противника

$\text{attacker}(m) \wedge \text{attacker}(pk) \Rightarrow \text{attacker}(\text{pencrypt}(m, pk))$
 $\text{attacker}(sk) \Rightarrow \text{attacker}(pk(sk))$
 $\text{attacker}(\text{pencrypt}(m, pk(sk))) \wedge \text{attacker}(sk) \Rightarrow \text{attacker}(m)$
 $\text{attacker}(m) \wedge \text{attacker}(sk) \Rightarrow \text{attacker}(\text{sign}(m, sk))$
 $\text{attacker}(\text{sign}(m, sk)) \Rightarrow \text{attacker}(m)$
 $\text{attacker}(\text{sign}(m, sk)) \wedge \text{attacker}(pk(sk)) \Rightarrow \text{attacker}(m)$
 $\text{attacker}(m) \wedge \text{attacker}(k) \Rightarrow \text{attacker}(\text{sencrypt}(m, k))$
 $\text{attacker}(\text{sencrypt}(m, k)) \wedge \text{attacker}(k) \Rightarrow \text{attacker}(m)$
 $\text{attacker}(a[])$

Выводимо ли событие $\text{attacker}(s)$?

Пример 1: выводимость события $attacker(s)$

Последовательность применения дизъюнктов Хорна, приводящая к нарушению секретности s

- ① $attacker(a[])$
- ② $attacker(a[]) \Rightarrow attacker(pk(a[]))$
- ③ $attacker(pencrypt(sign(k[pk(a[])], skA[]), pk(a[]))) \wedge attacker(a[]) \Rightarrow attacker(sign(k[pk(a[])], skA[]))$
- ④ $attacker(sign(k[pk(a[])], skA[])) \wedge attacker(pk(skB[])) \Rightarrow attacker(k[pk(a[])])$
- ⑤ $attacker(sencrypt(s, k[pk(a[])])) \wedge attacker(k[pk(a[])]) \Rightarrow attacker(s)$

Протокол Диффи-Хеллмана Proverif

- (pr_C, pk_C) – долговременные (закрытый, открытый) ключи клиента
- (x, g^x) – сессионные (закрытый, открытый) ключи клиента
- (pr_S, pk_S) – долговременные (закрытый, открытый) ключи сервера
- (y, g^y) – сессионные (закрытый, открытый) ключи сервера

Диффи-Хеллман

Client (pkC, privC)		Server (pkS, privS)
- генерация g^x	$\langle pkC, g^x \rangle$ ----->	
	$\langle pkS, g^y \rangle$ <-----	- генерация g^y
- $client_key = g^{y^x}$		- $server_key = g^{x^y}$
$key = g^{(x*y)}$ -- общий секрет		

Протокол Диффи-Хеллмана Proverif

- (pr_C, pk_C) – долговременные (закрытый, открытый) ключи клиента
- (x, g^x) – сессионные (закрытый, открытый) ключи клиента
- (pr_S, pk_S) – долговременные (закрытый, открытый) ключи сервера
- (y, g^y) – сессионные (закрытый, открытый) ключи сервера

Диффи-Хеллман

Client ($pk_C, priv_C$)		Server ($pk_S, priv_S$)
- генерация g^x	$\langle pk_C, g^x \rangle$ ----->	
	$\langle pk_S, g^y \rangle$ <-----	- генерация g^y
- $client_key = g^{y^x}$		- $server_key = g^{x^y}$
$key = g^{(x*y)}$ -- общий секрет		

Противник отправляет серверу сообщение $\langle pk_E, g^x \rangle$ от лица клиента



Протокол Диффи-Хэллмана не является стойким по отношению к активному противнику

Proverif: пример описания протокола BADH

- (pr_C, pk_C) – долговременные (закрытый, открытый) ключи клиента
- (x, g^x) – сессионные (закрытый, открытый) ключи клиента
- $sig_C(mess)$ – ЭЦП на закрытом ключе клиента
- (pr_S, pk_S) – долговременные (закрытый, открытый) ключи сервера
- (y, g^y) – сессионные (закрытый, открытый) ключи сервера
- $sig_S(mess)$ – ЭЦП на закрытом ключе сервера

Client (pkC, privC)		Server (pkS, privS)
- генерация g^x	$\langle g^x \rangle$ ----->	
	$\langle g^y, pk_S, sig_S(g^x, g^y) \rangle$ <-----	- генерация g^y
- $client_key = g^{xy}$	$\langle pk_C, sig_C(g^y, g^x) \rangle$ ----->	- $server_key = g^{xy}$
$key = g^{(x*y)}$ -- общий секрет		

Proverif: пример описания протокола BADH (нестойкость)

Client (pkC, privC)		Server (pkS, privS)
- генерация g^x	$\langle g^x \rangle$ ----->	
	$\langle g^y, pkS, sig_S(g^x, g^y) \rangle$ ----->	- генерация g^y
- $client_key = g^{xy}$	$\langle pkC, sig_C(g^y, g^x) \rangle$ ----->	- $server_key = g^{xy}$
$key = g^{(x*y)}$ -- общий секрет		

Противник отправляет серверу сообщение $\langle pk_E, sig_E(g^y, g^x) \rangle$ от лица клиента



Протокол не обеспечивает аутентификацию клиента перед сервером

Практическое задание

Необходимо построить модель на языке Tamarin протокола **ISO** и прислать её на почту vinevg2015@gmail.com или в телеграмм до 27.11.2020

- (pr_C, pk_C) – долговременные (закрытый, открытый) ключи клиента
- (x, g^x) – сессионные (закрытый, открытый) ключи клиента
- $sig_C(mess)$ – ЭЦП на закрытом ключе клиента
- (pr_S, pk_S) – долговременные (закрытый, открытый) ключи сервера
- (y, g^y) – сессионные (закрытый, открытый) ключи сервера
- $sig_S(mess)$ – ЭЦП на закрытом ключе сервера

Client (pk_C , $priv_C$)		Server (pk_S , $priv_S$)
- генерация g^x	$\langle pk_C, g^x \rangle$ ----->	
	$\langle pk_S, g^y, sig_S(g^x, g^y, pk_C) \rangle$ <-----	- генерация g^y
- $client_key = g^{y^x}$	$\langle sig_C(g^y, g^x, pk_S) \rangle$ ----->	- <u>$server_key = g^{x^y}$</u>
$key = g^{(x*y)}$ -- общий секрет		

Верификация на основе солвера cryptoverif

Криптографические примитивы (подпись, хэш, шифрование) – вероятностные процессы

- Безопасность криптографических примитивов обычно определяется как игра, в которую играют противник и “честные агенты”
- И противник, и “честные агенты” – вероятностные процессы, которые взаимодействуют друг с другом
- Безопасность означает, что для каждого полиномиального противника вероятность наступления события S пренебрежимо мала
- Игра G – вероятностный процесс, оперирующий действиями над криптографическими примитивами
- Задача – посчитать вероятность выпадения события S в игре G

Game 0 – вероятностный процесс (игра), выпадение события $S = S_0$ ($Pr(G_0 \rightarrow S_0)$) в которой есть нарушение свойства безопасности протокола

- 1 Строится последовательность игр *Game 0*, *Game 1*, ..., *Game n*, где S_0, S_1, \dots, S_n – такие, что:
 - Все игры определены над одним и тем же вероятностным пространством
 - $|Pr(G_i \rightarrow S_i) - Pr(G_{i-1} \rightarrow S_{i-1})| \leq \varepsilon_i$
 - $Pr(G_n \rightarrow S_n) = 0$
- 2 Тогда $Pr(G_0 \rightarrow S) \leq \varepsilon_0 + \dots \varepsilon_{n-1}$

Переход между играми $Game\ i$ и $Game\ i + 1$

- Игры $Game\ i$ и $Game\ i + 1$ определены над одним и тем же вероятностным пространством
- Игры $Game\ i$ и $Game\ i + 1$ действуют одинаково, если только не произойдет определенное “событие сбоя” F

Тогда $S_i \wedge \neg F \iff S_{i+1} \wedge \neg F$

Lemma 1 (Difference Lemma). *Let A, B, F be events defined in some probability distribution, and suppose that $A \wedge \neg F \iff B \wedge \neg F$. Then $|\Pr[A] - \Pr[B]| \leq \Pr[F]$.*

Proof. This is a simple calculation. We have

$$\begin{aligned} |\Pr[A] - \Pr[B]| &= |\Pr[A \wedge F] + \Pr[A \wedge \neg F] - \Pr[B \wedge F] - \Pr[B \wedge \neg F]| \\ &= |\Pr[A \wedge F] - \Pr[B \wedge F]| \\ &\leq \Pr[F]. \end{aligned}$$

The second equality follows from the assumption that $A \wedge \neg F \iff B \wedge \neg F$, and so in particular, $\Pr[A \wedge \neg F] = \Pr[B \wedge \neg F]$. The final inequality follows from the fact that both $\Pr[A \wedge F]$ and $\Pr[B \wedge F]$ are numbers between 0 and $\Pr[F]$. \square

Pseudo-Random Functions

- $\ell_1, \ell_2 \in \mathcal{N}$
- $\mathcal{F} = \{F_s\}_{s \in S}$ – семейство ключевых функций, где каждая функция F_s отображает $\{0, 1\}^{\ell_1}$ в $\{0, 1\}^{\ell_2}$
- Γ_{ℓ_1, ℓ_2} – множество всех функций, отображающих $\{0, 1\}^{\ell_1}$ в $\{0, 1\}^{\ell_2}$

PRF-преимущество \mathcal{A} равно

$$|Pr[s \xleftarrow{U} S : A^{F_s}() = 1] - Pr[f \xleftarrow{U} \Gamma_{\ell_1, \ell_2} : A^f() = 1]|$$

- $\ell \in \mathcal{N}$ и $\ell > \ell$
- $\mathcal{H} = \{H_k\}_{k \in K}$
- $|Pr[k \xleftarrow{U} K : H_k(w) = H_k(w') \mid w \neq w']| \leq \varepsilon_{uh}$

$$\mathcal{F}' = \{F'_{k,s}\}_{(k,s) \in (K \times S)}, \text{ где } F'_{k,s} = F_s(H_k(w))$$

! Если \mathcal{F} – псевдо-случайные, то и \mathcal{F}' – псевдо-случайные

Pseudo-Random Functions (2)

$k \xleftarrow{\$} K, s \xleftarrow{\$} S$
 $r \xleftarrow{\$} R$
for $i \leftarrow 1 \dots q$ do
 $w_i \leftarrow A(r, y_1, \dots, y_{i-1}) \in \{0, 1\}^\ell$
 $x_i \leftarrow H_k(w_i) \in \{0, 1\}^{\ell_1}$
 $y_i \leftarrow F_s(x_i) \in \{0, 1\}^{\ell_2}$
 $b \leftarrow A(r, y_1, \dots, y_q) \in \{0, 1\}$
output b

Game 0

$k \xleftarrow{\$} K, \boxed{f \xleftarrow{\$} \Gamma_{\ell_1, \ell_2}}$
 $r \xleftarrow{\$} R$
for $i \leftarrow 1 \dots q$ do
 $w_i \leftarrow A(r, y_1, \dots, y_{i-1}) \in \{0, 1\}^\ell$
 $x_i \leftarrow H_k(w_i) \in \{0, 1\}^{\ell_1}$
 $\boxed{y_i \leftarrow f(x_i) \in \{0, 1\}^{\ell_2}}$
 $b \leftarrow A(r, y_1, \dots, y_q) \in \{0, 1\}$
output b

Game 1

$$|Pr(S_0) - Pr(S_1)| \leq \varepsilon_{prf}$$

Pseudo-Random Functions (3)

```

 $k \xleftarrow{\$} K, Y_1, \dots, Y_q \xleftarrow{\$} \{0, 1\}^{\ell_2}$ 
 $r \xleftarrow{\$} R$ 
for  $i \leftarrow 1 \dots q$  do
     $w_i \leftarrow A(r, y_1, \dots, y_{i-1}) \in \{0, 1\}^{\ell}$ 
     $x_i \leftarrow H_k(w_i) \in \{0, 1\}^{\ell_1}$ 
    if  $x_i = x_j$  for some  $j < i$  then  $y_i \leftarrow y_j$  else  $y_i \leftarrow Y_i$ 
 $b \leftarrow A(r, y_1, \dots, y_q) \in \{0, 1\}$ 
output  $b$ 
    
```

Game 2

$$Pr(S_2) = Pr(S_1)$$

- F – событие в Game 3, $x_i = x_j$ для $i \neq j$
- $S_2 \wedge \neg F \iff S_3 \wedge \neg F$

$$|Pr(S_2) - Pr(S_3)| \leq Pr(F) \leq \varepsilon_{uh} \cdot \frac{q^2}{2}$$



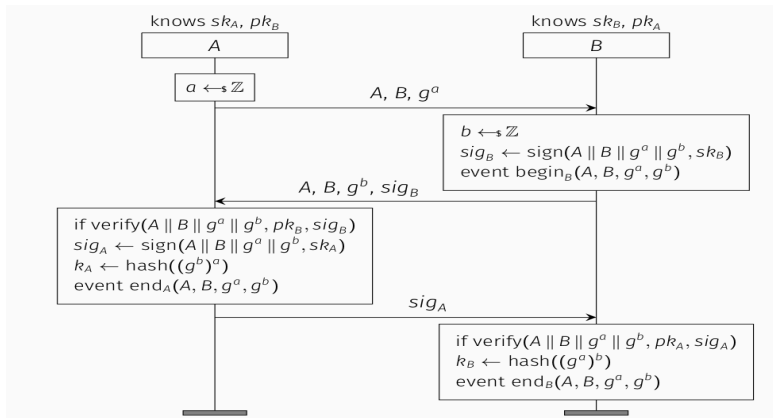
$$Pr(S_0) \leq \varepsilon_{prf} + \varepsilon_{uh} \cdot \frac{q^2}{2}$$

```

 $k \xleftarrow{\$} K, Y_1, \dots, Y_q \xleftarrow{\$} \{0, 1\}^{\ell_2}$ 
 $r \xleftarrow{\$} R$ 
for  $i \leftarrow 1 \dots q$  do
     $w_i \leftarrow A(r, y_1, \dots, y_{i-1}) \in \{0, 1\}^{\ell}$ 
     $x_i \leftarrow H_k(w_i) \in \{0, 1\}^{\ell_1}$ 
     $y_i \leftarrow Y_i$ 
 $b \leftarrow A(r, y_1, \dots, y_q) \in \{0, 1\}$ 
output  $b$ 
    
```

Game 3

Signed DH



Оценить вероятность нарушения свойства

$$\text{query } y : G, x : G; \text{inj-event}(\text{end}_A(A, B, x, y)) \implies \text{inj-event}(\text{begin}_B(A, B, x, y))$$

Formal View

- **Модель атаки:** возможность использовать любые правила вывода, представленные в теории
- **Угроза:** вывод специального события *bad*
- **Вычислительные ресурсы:** Не определены в явном виде

Computational View

- **Модель атаки:** возможность использовать все доступные оракулы
- **Угроза:** преимущество противника выше определённого порога
- **Вычислительные ресурсы:** количество запросов к внутренним алгоритмам (оракулам) полиномиально

Соотношение моделей противника при Formal View и Computational View

Модель Computational View сильнее модели Formal View

- Если протокол НЕ стойкий при Formal View, то он НЕ стойкий и при Computational View
- Если протокол стойкий при Computational View, то он стойкий и при Formal View

Однако

- если протокол стойкий при Formal View, то нет гарантий стойкости при Computational View
- если протокол НЕ стойкий при Computational View, то нет гарантий стойкости при Formal View

Доказательство НЕ стойкости протокола при Formal View конструктивно, т.е. **алгоритм возвращает атаку**