конференция
РусКрипто

Ежегодная международная научно-практическая конференция

«РусКрипто'2022»

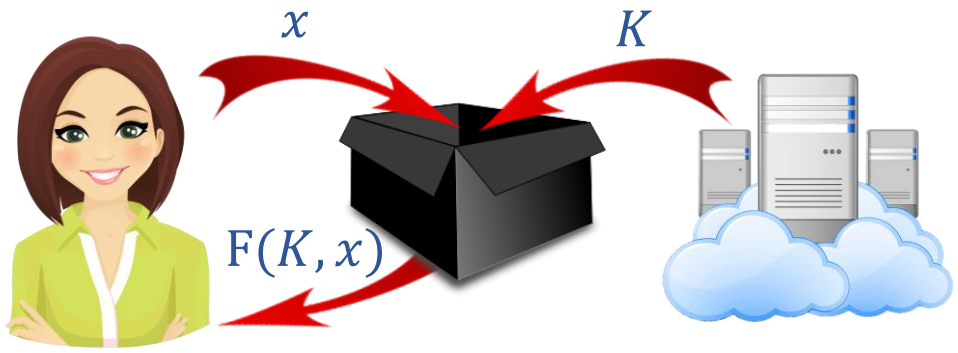
Псевдослучайные функции «с забыванием» в механизмах защиты на основе паролей

Ахметзянова Л.Р., зам. начальника отдела криптографических исследований, КриптоПро
Никифорова Л.О., инженер-аналитик, КриптоПро

OPRF. Определение

- OPRF – интерактивный протокол между клиентом и сервером.
  - Вход: клиент обладает данными $x \in \mathcal{X}$, сервер обладает ключом $K \in \mathcal{K}$.
  - Выход: клиент получает значение $\mathrm{F}(K, x) \in \mathcal{Y}$, сервер ничего не получает;

$\mathrm{F}: \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ – псевдослучайная функция.

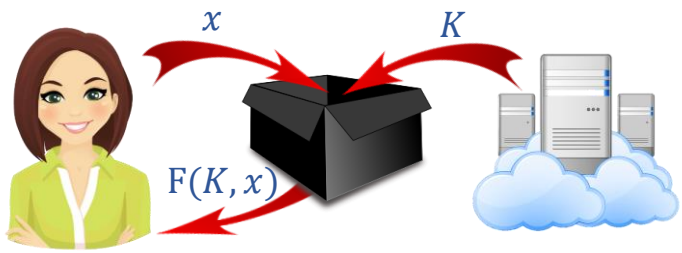


Freedman M. J. et al. Keyword search and oblivious pseudorandom functions //Theory of Cryptography Conference. – Springer, Berlin, Heidelberg, 2005. – С. 303-324.

Свойства (неформально)

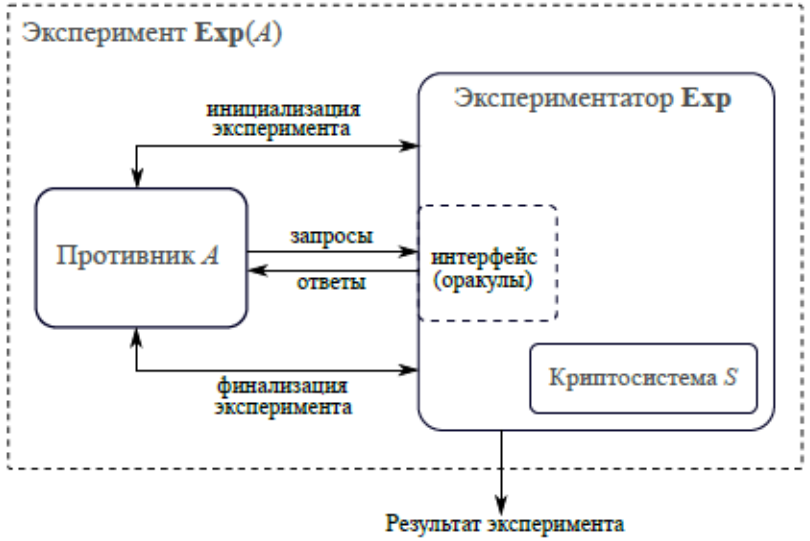
1. Конфиденциальность
   Сервер не должен получить никакой информации о x.

2. Свойство PRF
   До начала взаимодействия с сервером значение $\mathrm{F}(K, x)$ должно быть непредсказуемым для клиента.

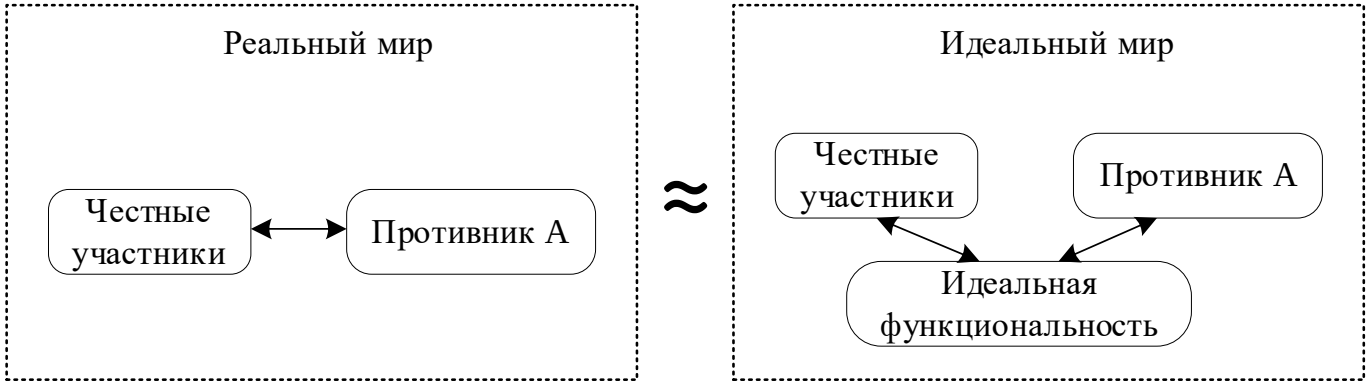


Подходы к формализации свойств

Алгоритмический (Game-based)

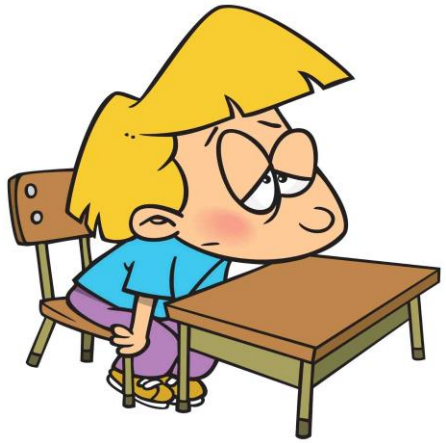


На основе идеальной функциональности (Simulation-based)



Алгоритмический подход

- Существуют только слабые модели.
- Рассматриваются только пассивные противники.



Casacuberta S., Hesse J., Lehmann A. SoK: Oblivious Pseudorandom Functions //Cryptology ePrint Archive. – 2022.
Lehmann A. ScrambleDB: Oblivious (Chameleon) Pseudonymization-as-a-Service //Proc. Priv. Enhancing Technol. – 2019. – Т. 2019. – №. 3. – С. 289-309.

Идеальная функциональность для OPRF

- Необходимо описать функционирование «идеального» протокола OPRF.
- Противник может компрометировать любого честного участника.
- OPRF выполняет свойство конфиденциальности и свойство PRF, если в реальном мире противник может получить только те данные, которые он может получить в идеальном.

Формальное описание идеальной функциональности

$F_S(x) \xleftarrow{u} \{0,1\}^\lambda$ для тех x, для которых F_S ещё не определена.

(Init): Доверенная сторона, получив Init от сервера S, отправляет (Init, S) противнику $\mathcal{A}$.

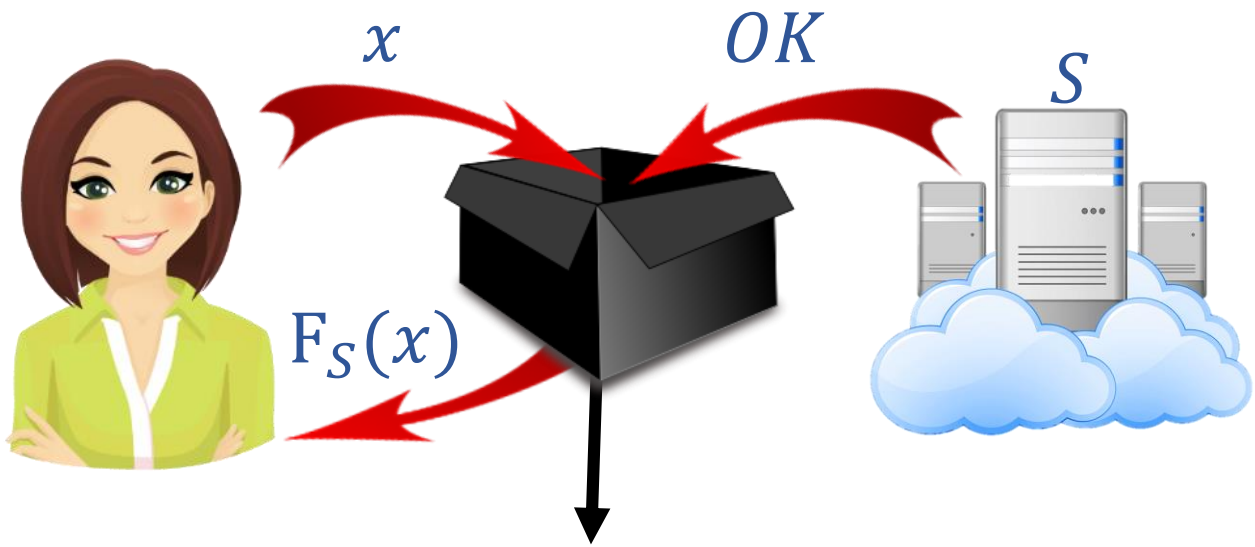
(Eval): Доверенная сторона, получив $(\text{Eval}, ssid, x)$ от клиента C:
- отправляет $(\text{Eval}, ssid, S)$ серверу S и противнику $\mathcal{A}$;
- сохраняет $(C, ssid, S, x)$.

(Procced): Доверенная сторона, получив $(\text{Procced}, ssid)$ от сервера S:
- отправляет $\mathcal{A}$ $(\text{Procced}, ssid)$ и получает в ответ $(ssid, S')$;
- возвращает ошибку, если $S \neq S'$ и сервер честный;
- возвращает $(ssid, F_{S'}(x))$ клиенту C.

(OfflineEval): Доверенная сторона, получив $(\text{OfflineEval}, x)$ от $\mathcal{A}$, отправляет в ответ $F_S(x)$, если сервер S скомпрометирован.

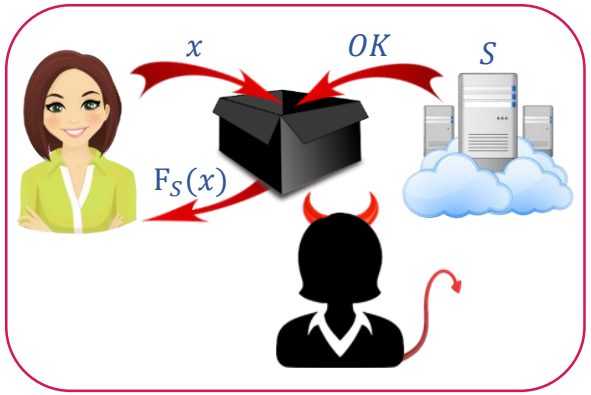
Casacuberta S., Hesse J., Lehmann A. SoK: Oblivious Pseudorandom Functions //Cryptology ePrint Archive. – 2022.

Принцип работы идеального протокола OPRF

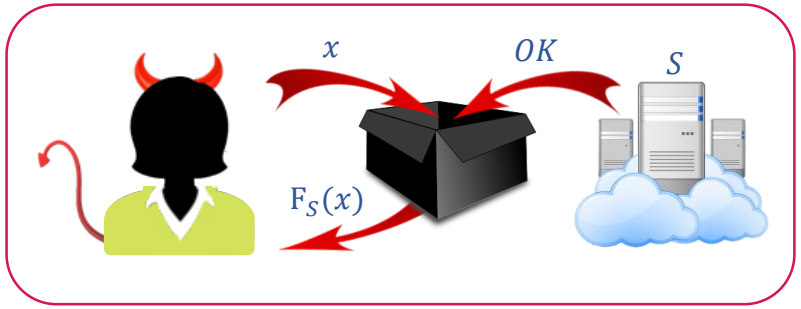


	x_1	...	x	...	x_n
S_1	$\mathrm{F}_{S_1}(x_1)$				$\mathrm{F}_{S_1}(x_n)$
...					
S			$\mathrm{F}_S(x)$		
...					
S_m	$\mathrm{F}_{S_m}(x_1)$				$\mathrm{F}_{S_m}(x_n)$

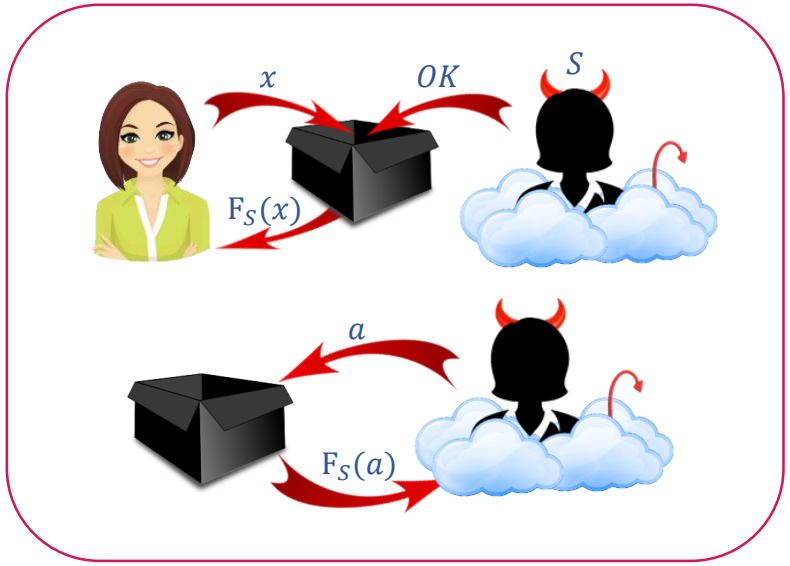
Влияние противника на идеальный протокол



Внешний противник:
не знает x;
не знает $\mathrm{F}_S(x)$.



Клиент-противник:
знает x;
знает $\mathrm{F}_S(x)$ – случайное значение.

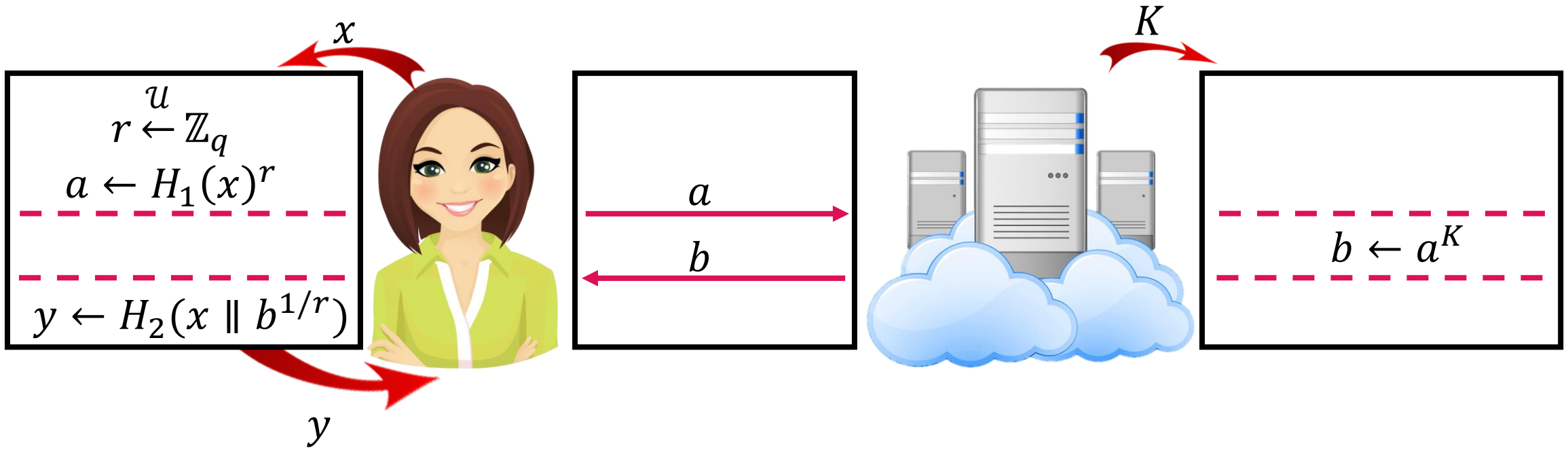


Сервер-противник:
не знает x;
есть доступ к оракулу $\mathrm{F}_S(\cdot)$.

OPRF. Пример

$\mathbb{G} = \langle g \rangle$ – циклическая группа порядка q;

$H_1: \mathcal{X} \to \mathbb{G}, \quad H_2: \mathbb{G} \to \mathcal{Y}$ – хэш-функции.



1. Конфиденциальность: за счет маскирования с помощью случайного r.
2. Свойство PRF: за счет сложности задачи one-more Gap DH.

Jarecki S., Kiayias A., Krawczyk H. Round-optimal password-protected secret sharing and T-PAKE in the password-only model //International Conference on the Theory and Application of Cryptology and Information Security. – Springer, Berlin, Heidelberg, 2014. – С. 233-253.

Применение OPRF

Зачем нужны эти свойства?

Их можно использовать в механизмах защиты на основе паролей.

Механизмы защиты на основе паролей

- ✓ Удобны в использовании.
- × Используют малоэнтропийные секреты.

<u>Решение проблемы:</u> использовать дополнительный сервер

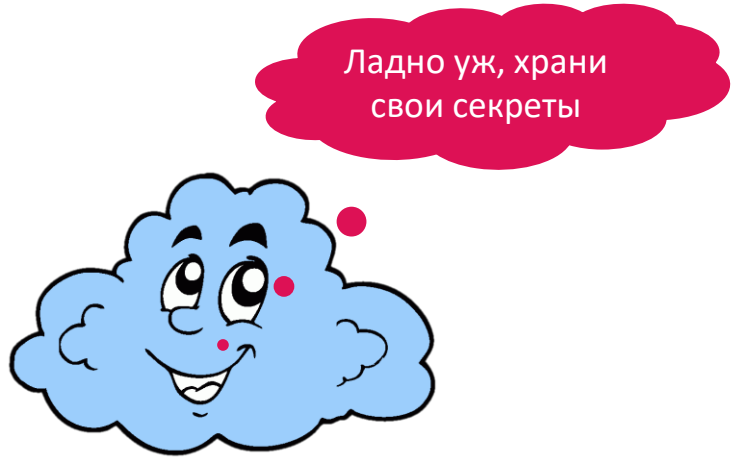
Механизмы защиты на основе паролей

- Менеджер паролей
- Хранение данных на устройстве

Менеджер паролей

Проблема: пользователю необходимо запоминать пароли от различных сервисов. Это приводит к тому, что используются слабые пароли, а часто и одинаковые для разных сервисов.

Подход к решению: использовать сервер для генерации паролей пользователя от различных сервисов и получения к ним доступа. Для доступа использовать один мастер-пароль.



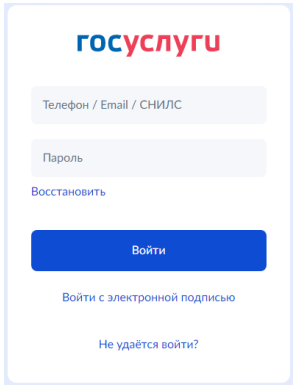
Менеджер паролей

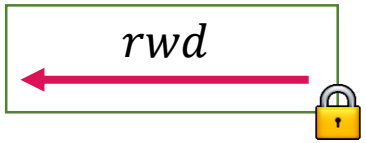
Что дает такой подход к решению:

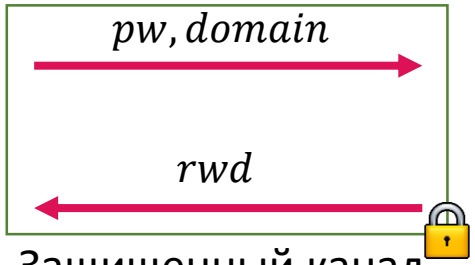
- для аутентификации на сервисе используется высокоэнтропийный секрет;
  - ✓ компрометация БД сервиса (хэш-значения паролей) не приводит к восстановлению секрета .
- для различных сервисов используются разные секреты;
  - ✓ один сервис не может аутентифицировать в качестве пользователя на другом сервисе.

! Что может сделать противник – менеджер паролей зависит от реализации подхода.

Классическое решение







Защищенный канал

Пароль	Ресурс
$E_{pw}(rwd)$	domain

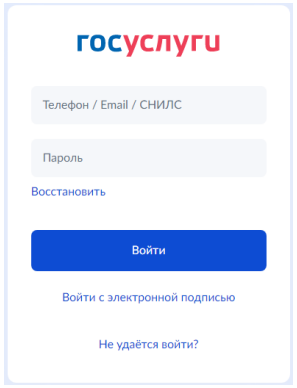
Противник – сервер менеджера паролей

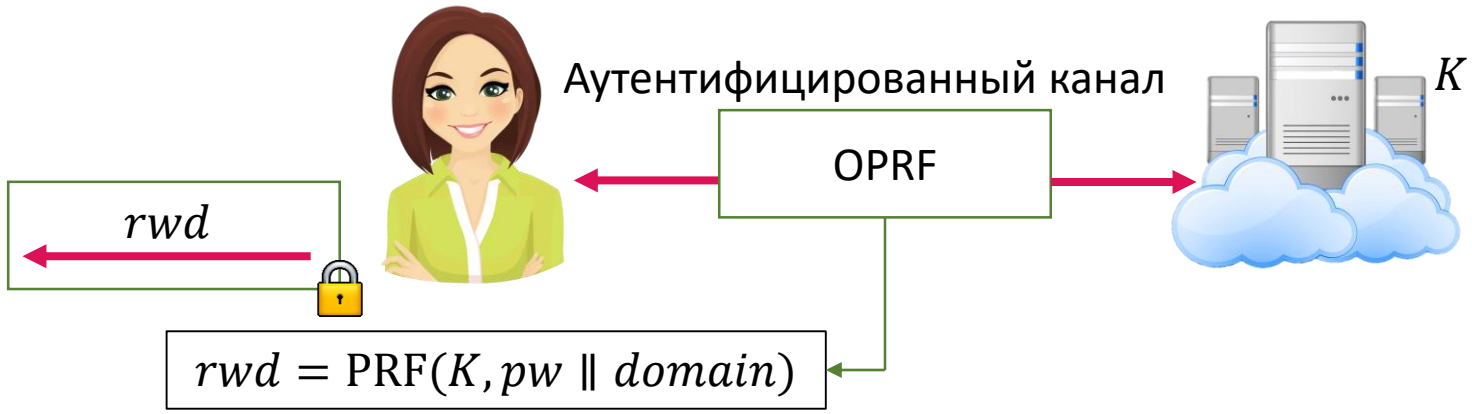
Получает мастер-пароль клиента.

Может расшифровать пароль клиента от любого сервиса.

Решение на основе OPRF

SPHINX





Противник – сервер менеджера паролей

Получает высокоэнтропийную часть секретов клиента от всех сервисов.

Необходимо выполнять онлайн перебор пароля pw.

Shirvanian M. et al. Building and studying a password store that perfectly hides passwords from itself //IEEE Transactions on Dependable and Secure Computing. – 2019. – Т. 16. – №. 5. – С. 770-782.

Хранение данных на устройстве

Проблема: необходимо обеспечить защиту данных, хранимых на устройстве, в случае кражи устройства.

Подход к решению: для защиты данных использовать пароль, который не хранится на устройстве.

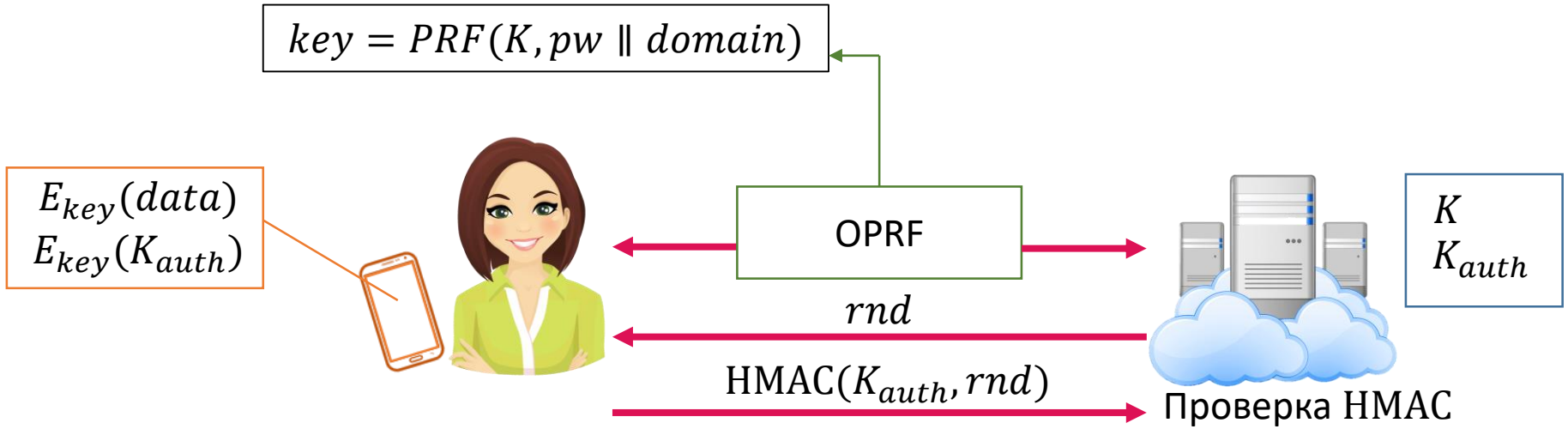
Классическое решение

Зашифровать данные на устройстве на пароле.

Противник – вор

Получает возможность оффлайн перебора пароля.

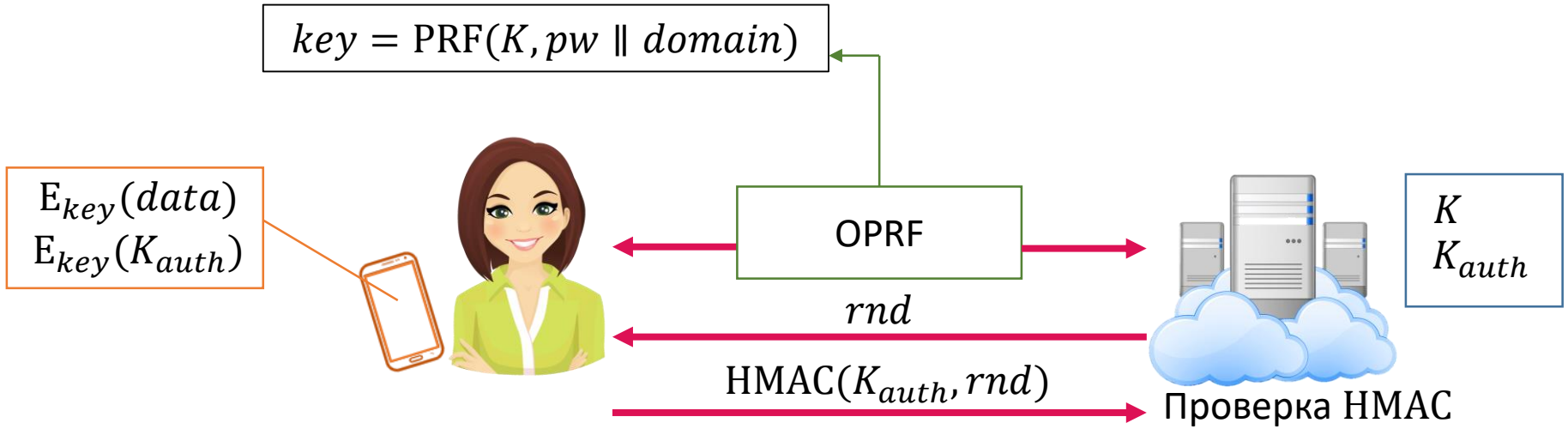
Решение на основе OPRF



Противник – вор

Необходимо выполнять онлайн перебор пароля pw.

Решение на основе OPRF



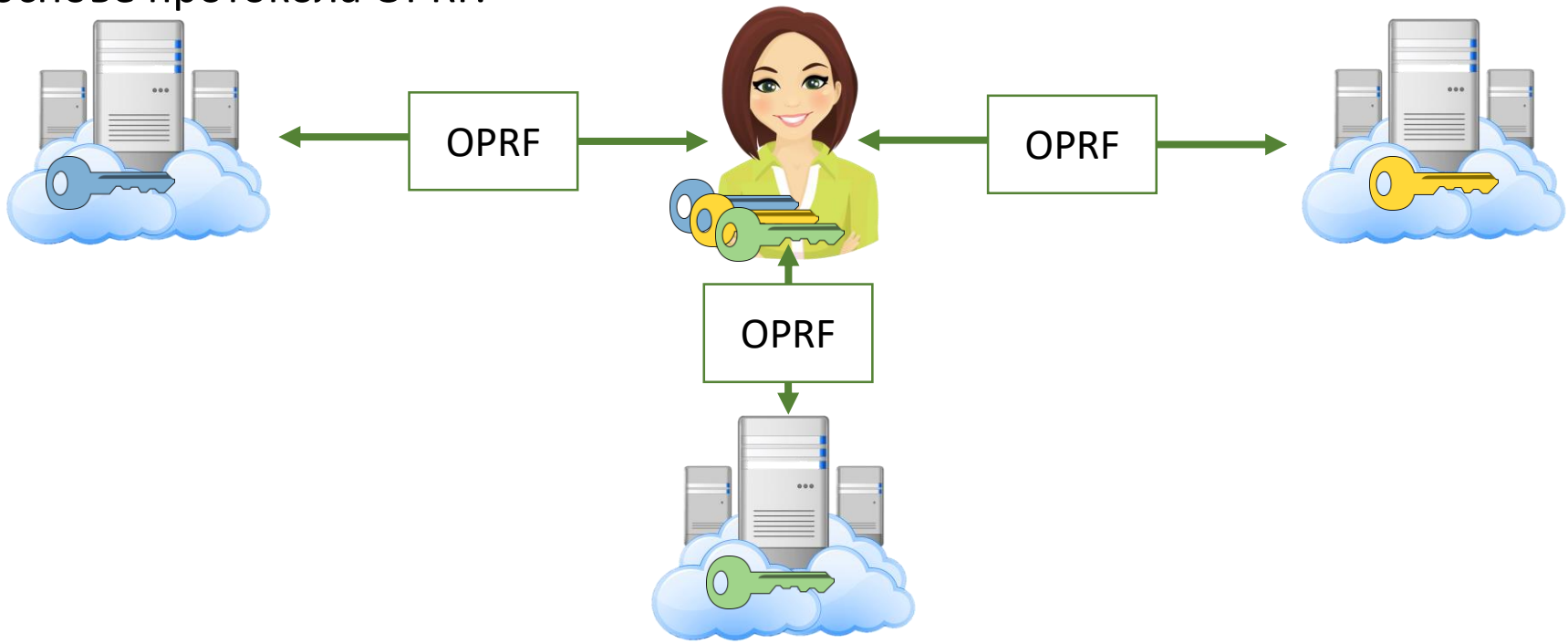
Необходимо выполнять оффлайн перебор пароля pw.

Противник – сервер + вор

Вместо заключения

Вместо одного сервера можно использовать несколько серверов для хранения секрета пользователя.

Для этого можно использовать схемы распределенного хранения с доступом по паролю на основе протокола OPRF.



Bagherzandi A. et al. Password-protected secret sharing //Proceedings of the 18th ACM conference on Computer and Communications Security. – 2011. – С. 433-444.

Спасибо за внимание!

Вопросы ???

Контактная информация

Электронная почта:

nikiforova@cryptopro.ru

Телефон:

+7 985 665-79-94

Сайт:

www.cryptopro.ru

One-more GAP DH

$\mathbb{G} = \langle g \rangle$ – циклическая группа;

Противник получает набор $(g, g^k, g_1, \dots, g_N)$, имеет доступ к оракулам:

- $(\cdot)^k$ (Q запросов);
- DDH.

Противник решает задачу one-more GAP DH, если возвращает Q+1 пару $\left\{\left(g_{t_i}, g_{t_i}^k\right)\right\}_{i=1..Q+1}$