УНИВЕРСИТЕТ ИТМО

ВПД

Виды контрольных сумм. Методы обеспечения целостности информации

Факультет Безопасности информационных технологий
Таранов Сергей Владимирович,
к.т.н., ординарный доцент Университета ИТМО
serg.tvc@gmail.com

Санкт-Петербург
2023

Виды контрольных сумм.
Как обнаружить и исправить ошибку

- **Методы теории кодирования**

* Линейные коды (могут быстро исправить и обнаружить ошибку, но их легко обмануть)

* Нелинейные коды (могут быстро обнаружить ошибку, обмануть сложнее, чем линейные, но возможно)

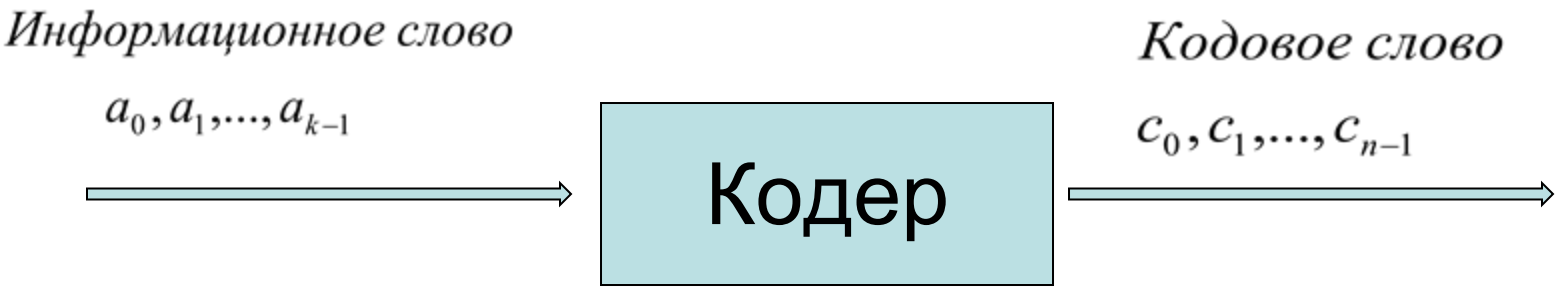
- **Методы криптографии**

*хэш функции (намного медленнее методов кодирования, могут только обнаружить ошибку, чтобы обмануть нужно подобрать коллизию)

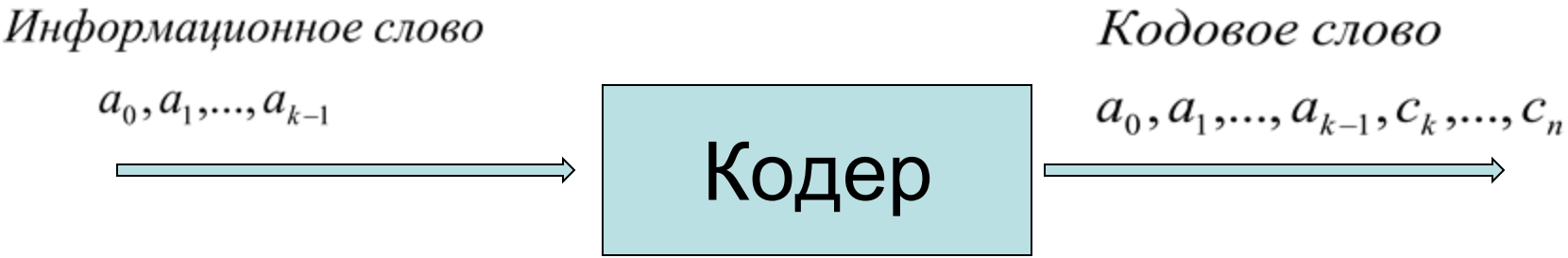
* Имитовставки (коды аутентификации) (аналогичны хэш функциями, однако, чтобы обмануть нужно еще и подобрать криптографический ключ)

* Подписи (медленнее по сравнению с хэш-функцией, обман не сложнее хэш-функции, есть возможности по подтверждению авторства)

Методы теории кодирования. Линейные коды



Систематическое кодирование



Кодирование - внесение избыточности

$$\left.\begin{aligned} c_0 &= a_0, \\ c_1 &= a_1, \\ &\dots \\ c_{k-1} &= a_{k-1}, \end{aligned}\right\} - \textit{информационные биты}$$

$$\left.\begin{aligned} c_k &= f_k(c_0,\dots,c_{k-1}), \\ &\dots \\ c_{n-1} &= f_{n-1}(c_0,\dots,c_{k-1}) \end{aligned}\right\} - \textit{проверочные биты},$$

$$f_k,\dots,f_{n-1} - \textit{линейные булевы функции}$$

Пример линейного систематического кодирования - добавление проверки на четность

Пример.

Информационное слово	Кодовое слово
000	0000
001	0011
010	0101
011	0110
100	1001
101	1010
110	1100
111	1111

$$c_0 = a_0,$$
$$c_1 = a_1,$$
$$c_2 = a_2,$$
$$c_3 = c_0 \oplus c_1 \oplus c_2.$$

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}$$

Порождающая матрица

Пусть γ - кодовое слово длины *n*

α - информационное слово длины *k*

$$\gamma = G \cdot \alpha$$

$$G = \begin{pmatrix} I_k \\ G_1 \end{pmatrix}$$

G – *nxk порождающая матрица* кода

Порождающая матрица

- *Пример.* $\gamma = G \cdot \alpha$

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \end{bmatrix} = \left(\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{array}\right) \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

- *Длина слов n=7, число иформационных разрядов =4, число проверочных разрядов n-k=3*

Проверки

- *Пример.* Получаем *проверки*

$$c_4 = c_0 \oplus c_2 \oplus c_3,$$
$$c_5 = c_0 \oplus c_1 \oplus c_2,$$
$$c_6 = c_1 \oplus c_2 \oplus c_3,$$

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \end{bmatrix} = \left(\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{array}\right) \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Проверочная матрица

- *Пример.*

$$\begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \end{bmatrix} = 0$$

$$c_0 \oplus c_2 \oplus c_3 \oplus c_4 = 0$$
$$c_0 \oplus c_1 \oplus c_2 \oplus c_5 = 0,$$
$$c_1 \oplus c_2 \oplus c_3 \oplus c_6 = 0,$$

- *H – (n-k)xn проверочная матрица:*

$$H\gamma = 0$$

Методы теории кодирования.

УНИВЕРСИТЕТ ИТМО

Методы теории кодирования отлично подходят для обнаружения случайных ошибок, которые возникают в результате сбоя работы аппаратного обеспечения, искажений при передаче сигнала через беспроводную среду и т.д.

А что если ошибка не имеет случайного характера?

- ошибка вносится злоумышленником;

ITMO more than a UNIVERSITY

Методы криптографии. Хэш-функция

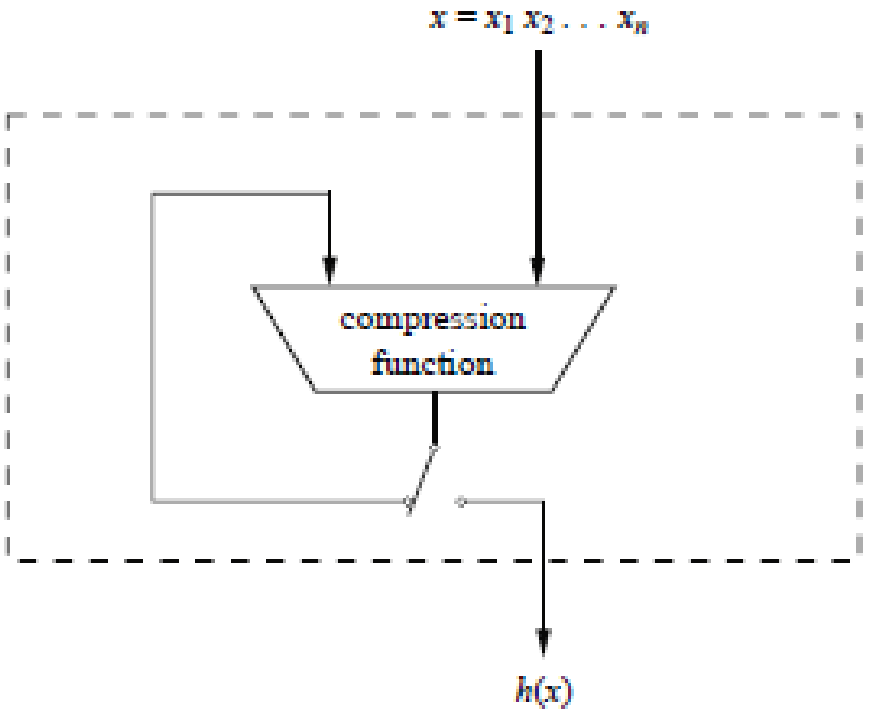
УНИВЕРСИТЕТ ИТМО

Хэш-функция h– функция, определенная на битовых строках произвольной длины со значениями в строках битов фиксированной длины. Ее значение называется *хэш-кодом*.

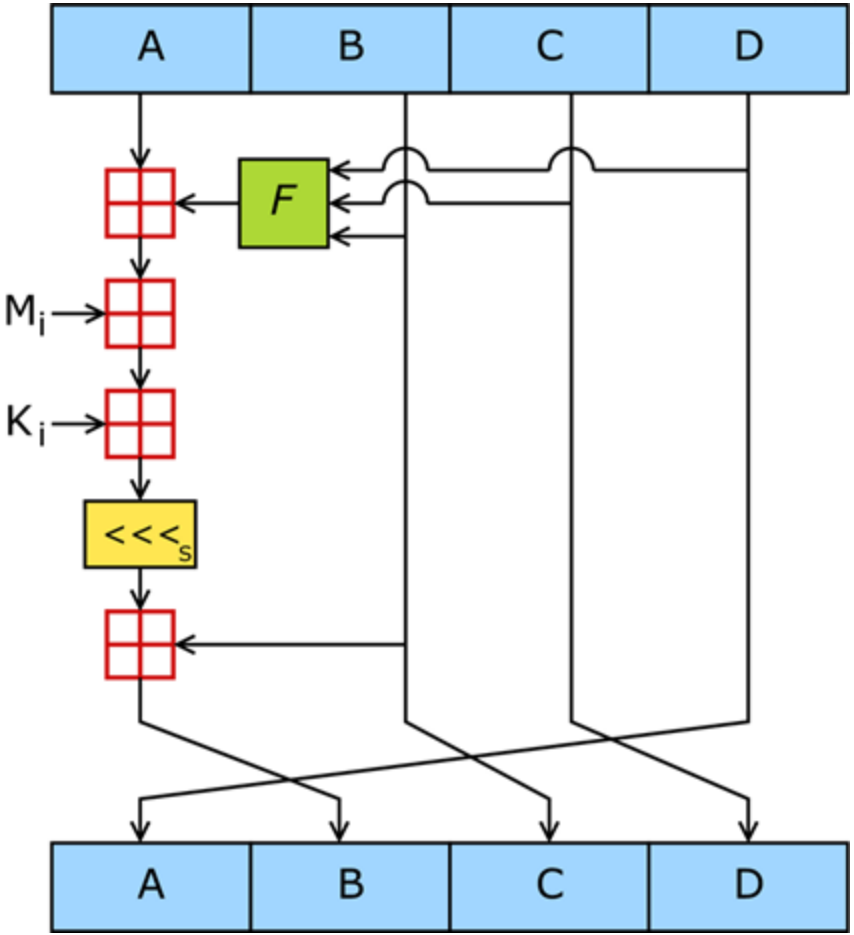
- Защищенность от восстановления прообразов: по Y из МЗХФ подобрать x из ОО: $h(x) = Y$
- Защищенность от повторений: не $\exists\, x \neq x' : h(x) = h(x')$
- Защищенность от вторых прообразов: по данному M невозможно найти $M' \neq M$: $h(M) = h(M')$.

ITMO more than a UNIVERSITY

Схема Меркла-Дамгарда



MD5.Структура



MD-5

После некоторой первоначальной обработки MD5 обрабатывает входной текст 512-битовыми блоками, разбитыми на 16 32-битовых подблоков. Выходом алгоритма является набор из четырех 32-битовых блоков, которые объединяются в единое 128-битовое хэш-значение.

Во первых, сообщение дополняется так, чтобы его длина была на 64 бита короче числа, кратного 512. Этим дополнением является 1, за которой вплоть до конца сообщения следует столько нулей, сколько нужно . Затем, к результату добавляется 64-битовое представление длины сообщения (истинной, до дополнения). Эти два действия служат для того, чтобы длина сообщения была кратна 512 битам (что требуется для оставшейся части алгоритма), и чтобы гарантировать, что разные сообщения не будут выглядеть одинаково после дополнения . Инициализируются четыре переменных:

A = 0x01234567

B = 0x89abcdef

C = 0xfedcba98

D = 0x76543210

Они называются **переменными сцепления**.

Раундовые функции MD5

$$F(X,Y,Z) = (X \wedge Y) \vee ((\neg X) \wedge Z)$$

$$G(X,Y,Z) = (X \wedge Z) \vee (Y \wedge (\neg Z))$$

$$H(X,Y,Z) = X \oplus Y \oplus Z$$

$$I(X,Y,Z) = Y \oplus (X \vee (\neg Z))$$

($\oplus$ - это XOR, $\wedge$ - AND, $\vee$ - OR, а $\neg$ - NOT.)

Парадокс дней рождений

в ящике находятся m шариков

Вероятность того, что

после n вытаскиваний нам попадется хотя бы два шарика одного цвета, равна

$$1 - \frac{m^{(n)}}{m^n},$$

где

$$m^{(n)} = m(m-1)(m-2)\cdots(m-n+1).$$

$$1 - \frac{365^{(23)}}{365^{23}} = 0,507.$$

Методы криптографии. Код аутентификации сообщения

MAC (имитовставка, message authentication code — код аутентичности сообщения) — контрольная сумма, которая добавляется к сообщению и предназначена для обеспечения его целостности и аутентификации источника данных.

MAC обычно применяется для обеспечения целостности и защиты от фальсификации передаваемой информации.

$$\mathrm{HMAC}_K(text) = \mathrm{H}\left((K_0 \oplus opad) \| \mathrm{H}\left((K_0 \oplus ipad) \| text\right)\right)$$

Методы криптографии. Цифровая подпись

Схема *подписи с приложением.*

СООБЩЕНИЕ + секретный ключ Алисы = ПОДПИСЬ

СООБЩЕНИЕ + ПОДПИСЬ + ОТКРЫТЫЙ КЛЮЧ АЛИСЫ = ДА/НЕТ

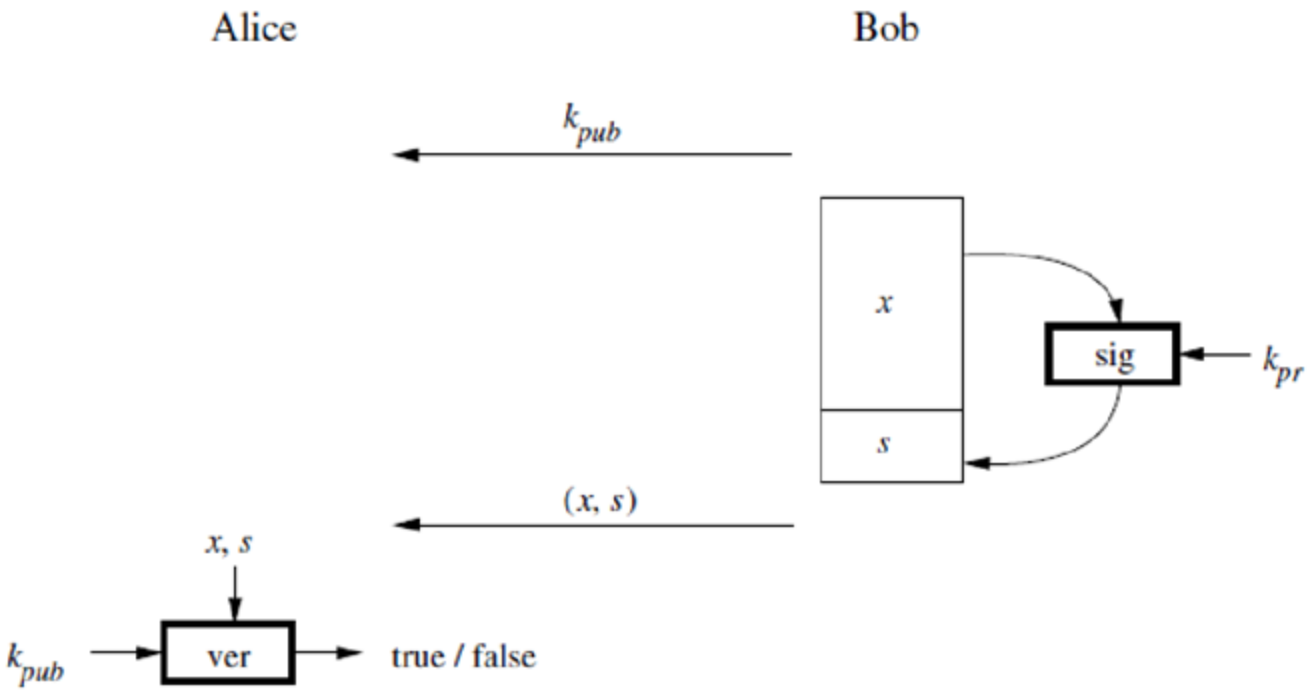
Схема подписи с *восстановлением сообщения.*

СООБЩЕНИЕ + секретный ключ Алисы = ПОДПИСЬ

ПОДПИСЬ + ОТКРЫТЫЙ КЛЮЧ АЛИСЫ = ДА/НЕТ + СООБЩЕНИЕ

Схема цифровой подписи.

- Секретное преобразование подписи s
- Открытое преобразование проверки *V*


Alice
Bob
k_{pub}
x
sig
k_{pr}
s
(x, s)
x, s
k_{pub}
ver
true / false

Свойства подписи

1. Подпись достоверна. Она убеждает получателя документа в том, что подписавший сознательно подп и-сал документ.
2. Подпись неподдельна. Она доказывает, что именно подписавший, и никто иной, сознательно подписал документ.
3. Подпись не может быть использована повторно. Она является частью документа, жулик не сможет перенести подпись на другой документ.
4. Подписанный документ нельзя изменить. После того, как документ подписан, его невозможно изм е-нить.
5. От подписи не возможно отречься. Подпись и документ материальны. Подписавший не сможет вп о-следствии утверждать, что он не подписывал документ.

Схема цифровой подписи RSA

RSA Keys

- Bob's private key: $k_{pr} = (d)$
- Bob's public key: $k_{pub} = (n, e)$

Basic RSA Digital Signature Protocol

Alice		Bob
		$k_{pr} = d,\ k_{pub} = (n, e)$
	$\xleftarrow{(n,e)}$	
		compute signature: $s = \text{sig}_{k_{pr}}(x) \equiv x^d \bmod n$
	$\xleftarrow{(x,s)}$	

verify: $\text{ver}_{k_{pub}}(x, s)$

$x' \equiv s^e \bmod n$

$x' \begin{cases} \equiv x \bmod n & \Longrightarrow \text{valid signature} \\ \not\equiv x \bmod n & \Longrightarrow \text{invalid signature} \end{cases}$

Existential Forgery Attack Against RSA Digital Signature

Alice **Oscar** **Bob**

$k_{pr} = d$
$k_{pub} = (n, e)$

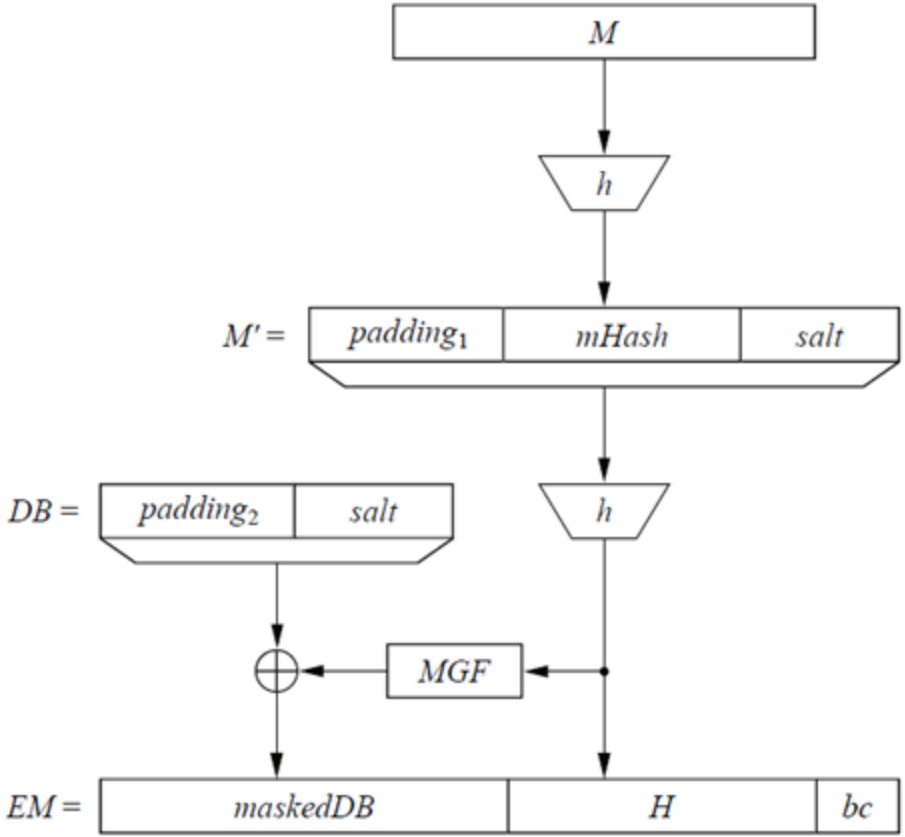
$\xleftarrow{(n,e)}$ $\xleftarrow{(n,e)}$

1. choose signature:
   $s \in \mathbb{Z}_n$
2. compute message:
   $x \equiv s^e \bmod n$

$\xleftarrow{(x,s)}$

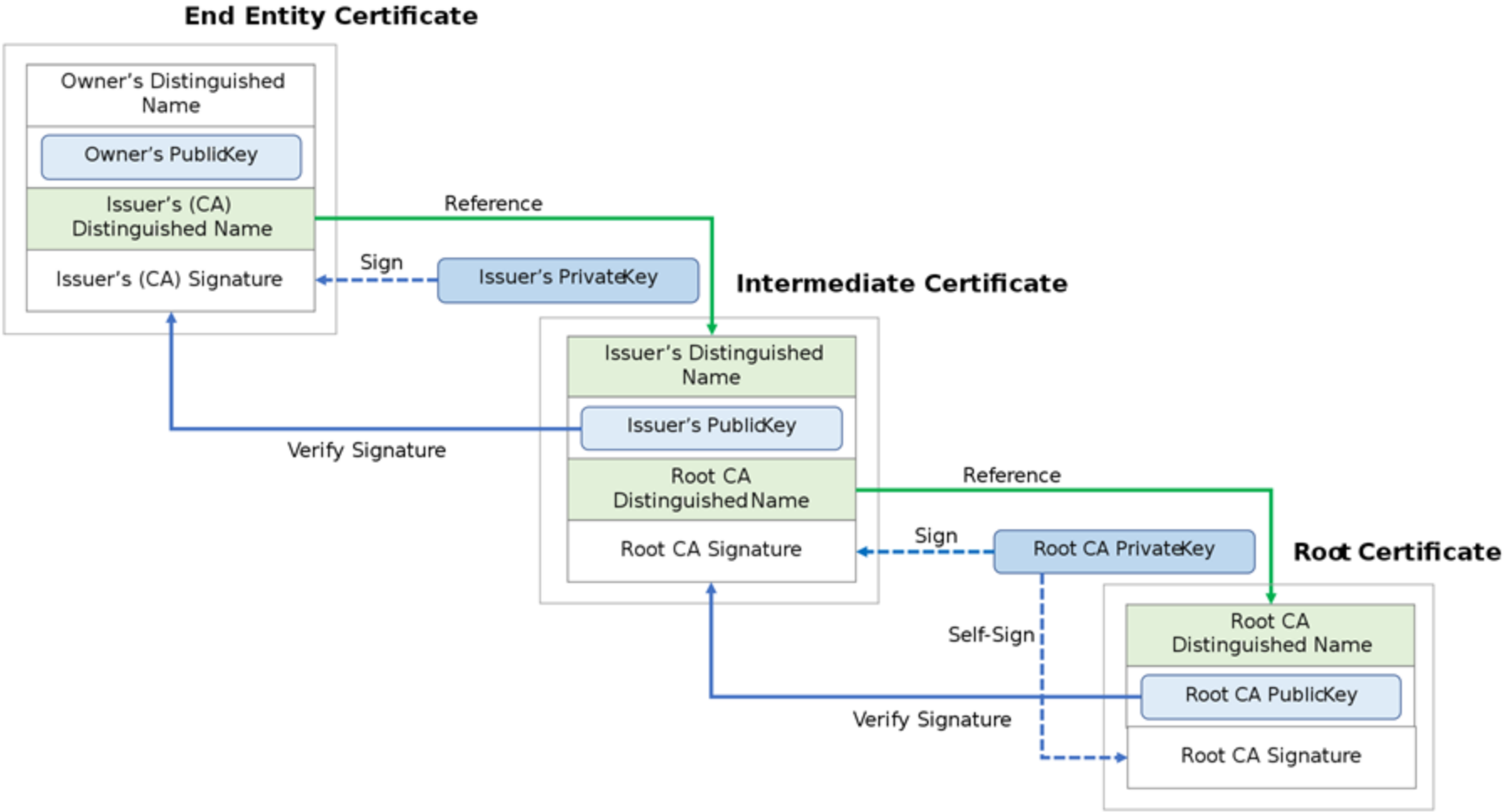
verification:
$s^e \equiv x' \bmod n$
since $x' = x$
$\Longrightarrow$ valid signature!

Алгоритмы дополнения и маскировки в подписи



Структура сертификата

- Сертификат
  - Версия
  - Серийный номер
  - Идентификатор алгоритма подписи
  - Имя издателя
  - Период действия:
    - Не ранее
    - Не позднее
  - Имя субъекта
  - Информация об открытом ключе субъекта:
    - Алгоритм открытого ключа
    - Открытый ключ субъекта
  - Уникальный идентификатор издателя
  - Уникальный идентификатор субъекта
  - Дополнения
- Алгоритм подписи сертификата
- Подпись сертификата (обязательно для всех версий)


End Entity Certificate
Owner's Distinguished Name
Owner's PublicKey
Issuer's (CA) Distinguished Name
Issuer's (CA) Signature
Reference
Sign
Issuer's PrivateKey
Intermediate Certificate
Issuer's Distinguished Name
Issuer's PublicKey
Root CA Distinguished Name
Root CA Signature
Verify Signature
Reference
Sign
Root CA PrivateKey
Root Certificate
Self-Sign
Root CA Distinguished Name
Root CA PublicKey
Root CA Signature
Verify Signature