

Rust Crypto For XChain

Rust Crypto For XChain

Xuper-sdk-go vs rust-sgx for Crypto

ECC

ECDSA

ECDSA签名:

ECDSA 验证签名

ECIES算法

加密

解密

证明过程

Refer

有限域运算

表示

多项式基表示法

正规基表示法

运算

加法

减法

乘法

除法

求逆

Xuper-sdk-go vs rust-sgx for Crypto

	超级链	crate-rust-sgx
ecdsa	crypto/ecdsa: P256-SHA256-ANS1	ring::P256-SHA256-ASN1 ring::P256-SHA384-ASN1
hash	crypto/hmac crypto/sha512 crypto/sha256 "golang.org/x/crypto/ripemd160"	ring::{hmac,sha256,sha512} ripemd160
encode	self/base58 自己实现的	base58
bigint	math/bigint	num-bigint
rand	crypto/rand	rand
aes	crypto/aes (Rijndael 128, 192, 256)	ring
ecies	Kylom's implementation curve: P256	Done
sign	multi_sign, schnorr_ring_sign, schnorr_sign	需要实现
hdwallet/keychain	hdwallet/keychain	需要实现

ECC

ECDSA

Parameter	
CURVE	the elliptic curve field and equation used
G	elliptic curve base point, a point on the curve that generates a subgroup of large prime order n
n	integer order of G , means that $n \times G = O$, where O is the identity element.
k	the private key (randomly selected)
P	the public key (calculated by elliptic curve)
M	the message to send

ECDSA签名:

$$\begin{aligned}P &= (x_1, y_1) = k \times G \\S &= k^{-1}(\text{Hash}(M) + k * x_1) \bmod p \\Signature &= (x_1, S)\end{aligned}$$

ECDSA 验证签名

$$\begin{aligned}P' &= S^{-1} * \text{Hash}(M) \times G + S^{-1} * x_1 \times P \\&= P\end{aligned}$$

- 证明

$$\begin{aligned}P' &= S^{-1} * \text{Hash}(M) \times G + S^{-1} * k \times G \\&= (S^{-1} * \text{Hash}(M) + S^{-1} * k) \times G \\&= (\text{Hash}(M) + x_1) * S^{-1} \times G \\&= (\text{Hash}(M) + x_1) * (k^{-1}(\text{Hash}(M) + k))^{-1} \times G \\&= (\text{Hash}(M) + x_1) * k * (\text{Hash}(M) + k)^{-1} \times G \\&= k \times G \\&= (x_1, y_1)\end{aligned}$$

ECIES算法

为了向Bob发送ECIES加密信息，Alice需要以下信息：

- 密码学套件（KDF，MAC，对称加密E）
- 椭圆曲线(p, a, b, G, n, h)

- Bob的公钥:

$$K_b, K_b = k_b G, k_b \in [1, n - 1]$$

- 共享信息

$$S_1, S_2$$

- 无穷远点O

加密

Alice使用Bob的公钥加密消息m:

$$\begin{aligned} & \text{For random } r \in [1, n - 1], \text{ calculate } R = rG \\ & \text{derive shared secret : } S = P_x, \text{ where } P = P(P_x, P_y) = rK_b, P \neq O \\ & \text{derive } K_E || K_M = KDF(S || S_1) \\ & \text{encrypt message } m : c = E(k_E; m) \\ & \text{calculate MAC : } d = MAC(k_M; c || S_2) \\ & \text{output : } R || c || d \end{aligned}$$

解密

Bob解密密文 R || c || d的步骤如下:

$$\begin{aligned} & \text{derive shared secret : } S = P_x, P = P(P_x, P_y) = k_B R \\ & \text{derive } K_E || K_M = KDF(S || S_1) \\ & \text{verify MAC : } d == MAC(k_M; c || S_2) \\ & \text{decrypt : } m = E^{-1}(k_E; c) \end{aligned}$$

证明过程

we need ensure S is really shared by Alice and Bob:

$$P = K_B r = k_B R$$

Refer

1. https://en.wikipedia.org/wiki/Integrated_Encryption_Scheme

有限域运算

表示

有限域:

$$GF(p^n)$$

这里专门针对p=2为特征的多项式进行计算。

多项式基表示法

正规基表示法

运算

加法

减法

乘法

除法

求逆