حوری دهش 9821413

سوال 1:

Key: 9821413

Plaintext: “Let’s put a smile on that face”

Key	9	8	2		1	4	1	3	9	8	2	1	4	1	3	9	8	2	1	4	1	3	9	8
Plaintext	l	e	t	'	s	p	u	t	a	s	m	i	l	e	o	n	t	h	a	t	f	a	c	e
Ciphertext	u	m	v	'	t	t	v	w	j	a	o	j	p	f	r	w	b	j	b	x	g	d	l	m

Ciphertext: "Umv'ttvwjaojpfrwbjbxgdlm"

سوال 2:

در حمله Only Ciphertext چون ما Ciphertext را داریم پس باید Ciphertext را سه بار رمزگشایی کنیم یعنی از هر سه کلید باید استفاده کنیم تا به plaintext برسیم پس پیچیدگی حمله جستجوی فراگیر در این حالت میشود $2^{56} * 2^{56} * 2^{56} = 2^{168}$ یعنی برای به دست اوردن سه کلید به 2^{168} زمان نیاز داریم

در حمله known plaintext چون ما plaintext را داریم، فقط از دو کلید استفاده میکنیم در نهایت پیچیدگی حمله جستجوی فراگیر برای پیدا کردن دو کلید میشود $2^{56} * 2^{56} = 2^{112}$

سوال 3:

در حالت CBC بلوک‌های رمزگشایی شده یکی پس از دیگری به عنوان ورودی به بلوک‌های بعدی داده می‌شوند. این بدین معنی است که اگر یک کلید یا یک IV به طور مجدد استفاده شود، تأثیرات آن بر بلوک‌های بعدی انتقال داده شده به سادگی قابل مشاهده خواهد بود. به عبارت دیگر اگر برای بلوک‌های بعدی از همان کلید و IV قبلی استفاده شود به دلیل تکرار شدن IV و کلید، در برخی از حالات امنیت الگوریتم CBC کاهش پیدا می‌کند یعنی اگر بلوک‌های متنی مشابه در ورودی وجود داشته باشد، توابع XOR و رمزنگاری انجام شده برای هر دو بلوک، مشابه خواهد بود و ممکن است با تحلیل الگوی برخی از بلوک‌ها حملاتی با موفقیت نفوذ کنند.

سوال 4:

SIK: (3,5,12,23)

n=47

m=6

M=1101

Public key:

3*6 mod 47 = 18

5*6 mod 47 = 30

12*6 mod 47 = 25

23*6 mod 47 = 44

(18,30,25,44)

Private key:

$m^{-1}\ mod\ n = \ 6^{-1}\ mod\ 47 = 8$

$m^{-1}m = 6m^{-1} = 1\ mod\ 47 \rightarrow \ m^{-1} = 8$

Ciphertext:

18+30+44 = 92

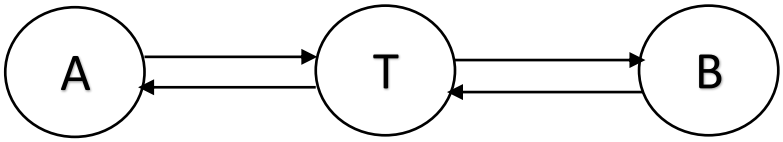
سوال 5:

حمله MITM به تبادل کلید Diffie-Hellman به رمزنگاری ECC زمانی رخ می دهد که یک مهاجم ارتباط بین دو طرف را رهگیری کرده و آن را تغییر دهد. در تبادل کلید ECC DH، آلیس و باب بر روی یک منحنی و یک نقطه پایه در منحنی توافق می کنند. هر کدام یک کلید خصوصی تولید کرده و از کلیدهای خصوصی خود و نقطه پایه برای تولید یک کلید عمومی استفاده می کنند. سپس طرفین کلید های عمومی خود را مبادله کرده و یک کلید مشترک را محاسبه می کنند که به عنوان کلید رمزگذاری متقارن استفاده می شود.

در حمله MITM به ECC DH، مهاجم ارتباط بین آلیس و باب را رهگیری می کند و هر یک از طرفین را به دیگری جعل می کند. مهاجم کلید عمومی خود را با استفاده از یک کلید خصوصی متفاوت تولید کرده و این کلید عمومی را طوری برای آلیس می فرستد که گویی کلید عمومی باب است. مهاجم همین کار را برای باب هم انجام می دهد و یک کلید عمومی متفاوت را به آلیس می فرستد که گویی کلید عمومی آلیس است.

وقتی آلیس و باب کلید مشترک خود را با استفاده از کلیدهای عمومی مهاجم محاسبه می‌کنند، مهاجم می‌تواند ارتباط بین آلیس و باب را رمزگشایی کرده بخواند و اصلاح کند که این کار را می تواند با

کلید خصوصی خودش انجام دهد و سپس آن را با کلید عمومی خودش دوباره رمزگذاری کند و ان را به گیرنده مورد نظر منتقل کند.



سوال 6:

Alice sends Bob: 12(2,7) = (153,36)

Bob sends Alice: 31(2,7) = (103,153)

Alice sends Bob: 12(103,153) = (137,54)

Bob sends Alice: 31(153,36) = (137,54)

Key: (137,54)

سوال 7:

n = ((9821413 mod 5) + 2) = 5

x	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

y	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1

z	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0

$m = maj(x_8, y_{10}, z_{10}) = (1, 0, 1) = 1$

x	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

y	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1

z	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0

$keystream1: 0\ xor\ 1\ xor\ 0 = 1$

$m = maj(x_8, y_{10}, z_{10}) = (0, 0, 1) = 0$

x	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

y	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1

z	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0

$keystream2: 1\ xor\ 1\ xor\ 0 = 0$

$m = maj(\ x_8\ , y_{10}\ , z_{10}\ ) = (\ 1\ , 1\ , 1) = 1$

x	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

y	1	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0

z	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0

$keystream3: 0\ xor\ 0\ xor\ 0 = 0$

$m = maj(\ x_8\ , y_{10}\ , z_{10}\ ) = (\ 0\ , 1\ , 1) = 1$

x	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

y	0	1	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0

z	1	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0

$keystream4: 0\ xor\ 0\ xor\ 0 = 0$

$m = maj(\ x_8\ , y_{10}\ , z_{10}\ ) = (\ 0\ , 0\ , 1) = 0$

x	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

y	1	0	1	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1

z	1	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0

$keystream5: 1\ xor\ 1\ xor\ 0 = 0$

$m = maj(\ x_8\ , y_{10}\ , z_{10}\ ) = (\ 1\ , 0\ , 1) = 1$

$key: 10000$

سوال 8:

ساختار گواهی X.509 شامل مجموعه ای از فیلدهای داده است که حاوی اطلاعاتی در مورد نهادی است که گواهی شناسایی می کند و همچنین اطلاعات مربوط به خود گواهی. مهم ترین فیلدهای داده در گواهی X.509 عبارتند از:

1- Version Number: این قسمت نشان دهنده نسخه استاندارد X.509 است که برای رمزگذاری گواهی استفاده شده است.

2- Serial Number: این فیلد حاوی یک شناسه منحصر به فرد است که توسط مرجع صدور گواهی (CA) اختصاص داده شده است تا گواهی را از سایر گواهی ها که صادر کرده است متمایز کند.

3- Signature Algorithm Identifier: این فیلد الگوریتم مورد استفاده توسط CA برای امضای گواهی را مشخص می کند، مانند RSA یا DSA.

4- Issuer Name: این فیلد حاوی نام CA است که گواهی را صادر کرده است.

5- Validity Period: این فیلد حاوی تاریخ هایی است که طی آن گواهینامه معتبر است از جمله تاریخ شروع و پایان.

6- Subject Name: این فیلد حاوی نام نهادی است که گواهینامه آن را شناسایی می کند مانند یک وب سایت یا یک فرد.

7- Subject Public Key Info: این فیلد حاوی اطلاعاتی درباره کلید عمومی مرتبط با موضوع است مانند الگوریتم مورد استفاده و خود کلید.

8- Certificate Extensions: این فیلد حاوی اطلاعات اضافی اختیاری درباره گواهی است مانند محدودیت‌های استفاده یا اطلاعات لغو گواهی.

9- Signature Value: این فیلد حاوی امضای دیجیتال گواهی است که با استفاده از کلید خصوصی CA ایجاد شده است.

هر یک از این فیلدهای داده با استفاده از یک نحو خاص که در استاندارد X.509 تعریف شده است کدگذاری می‌شوند و ساختار داده‌های حاصل به صورت دیجیتالی توسط CA امضا می‌شود تا از صحت آن اطمینان حاصل شود. ساختار گواهی X.509 می تواند بسیار پیچیده باشد، اما برای فعال کردن ارتباط امن از طریق اینترنت ضروری است.

سوال 9:

سیستم رمز ECC یکی از روش‌های رمزنگاری کلید عمومی است که بر پایه منحنی‌های بیضوی (elliptic curves) ساخته شده است. در این سیستم برای ایجاد جفت کلید عمومی و خصوصی، ابتدا یک منحنی بیضوی تعریف می‌شود این منحنی بیضوی یک نقطه مبدا و یک نقطه پایانی دارد که با نام "نقطه بین‌دو" شناخته می‌شود.

در این روش از یک الگوریتم رمزنگاری کلید عمومی استفاده شده که با استفاده از منحنی بیضوی برای هر کاربر یک جفت کلید عمومی و خصوصی تولید می‌شود. برای تولید کلید عمومی یک نقطه

اولیه در منحنی بیضوی انتخاب شده و بعد با تکراری از عملیات جمع نقطه‌ها، نقطه دیگری در منحنی بیضوی بدست می‌آید این نقطه جدید به عنوان کلید عمومی برای کاربر در نظر گرفته می‌شود.

برای تولید کلید خصوصی کاربر باید برای خود یک عدد رندوم انتخاب کند با استفاده از این عدد و نقطه عمومی که قبلاً برای کلید عمومی تولید شده بود بازیابی نقطه‌ای از منحنی بیضوی صورت می‌گیرد که به عنوان کلید خصوصی برای کاربر در نظر گرفته می‌شود.

برای ارسال یک پیام رمزنگاری شده به دستگاه گیرنده، پیام با استفاده از کلید عمومی گیرنده رمزنگاری می‌شود. برای رمزگشایی پیام، دستگاه گیرنده از کلید خصوصی خود استفاده می‌کند تا پیام رمزگشایی شده را باز کند.

سوال 10:

توضیحات در خود کد داده شده است

خروجی کد:

```
49 collisions found:
Message 1: 496690,  Hash 1: 474f2a6744d51503b5519c05f8c917f7
Message 2: 590703,  Hash 2: e03b9ccd3316164bbe6650b80b6ffbe0
Message 3: 672477,  Hash 3: a46c60ed21bbd1ec329999388ac419aa
Message 4: 553812,  Hash 4: dd15d97abfaea83f68f9dfdc6d8195de
Message 5: 954636,  Hash 5: ce8cdc536bc1726feaffa4c78e30a48b
Message 6: 648261,  Hash 6: acb60da99d0b187df851d31b88a4e440
Message 7: 775715,  Hash 7: 88d0c80d2cc407ee139324b207791819
Message 8: 291607,  Hash 8: cbf7f38508eea44e6ed5a64368c981f4
Message 9: 727654,  Hash 9: 181659421e27909cbe5651fc3654b2d1
Message 10: 830313,  Hash 10: 14567240df8445b1fc7b7437680ae2b8
Message 11: 885505,  Hash 11: 8f58ba8fd82829f6332e77594c66fb19
Message 12: 168643,  Hash 12: 362027ce05a53160a061e5c8788cef28
Message 13: 595267,  Hash 13: bbf8b7c22bf95165397bc69a2cc09683
Message 14: 972218,  Hash 14: 12d4902402bb40ebdaefb7532d64e6dc
Message 15: 751306,  Hash 15: 681278d36dc6b1cd68c2001e877e388d
Message 16: 448816,  Hash 16: f0ce003caea99b47886cfe2228fa2deb
Message 17: 78874,  Hash 17: f95b74d6de4feeaa269bc9032b196300
Message 18: 439893,  Hash 18: 110ba77f819e11303ff59009a6132b1a
Message 19: 316585,  Hash 19: 96ad1e5b5a49f241f186944127240311
Message 20: 624173,  Hash 20: a81ee84404b7f9906f45e1ea5fb5c1a8
Message 21: 866994,  Hash 21: 1004c0b27806e505f46b603e3e28cffe
Message 22: 191669,  Hash 22: 6afbb7587feb93bdb19e4978e02e4090
Message 23: 706472,  Hash 23: ede6e96b2bde3c529a460dc4593e8b48
Message 24: 282632,  Hash 24: 78469b233f3c68b57f48383ff9bd372a
```

```
Message 25: 285529,  Hash 25: 6c56c62eedee4ff07552b85d29539ec4
Message 26: 968444,  Hash 26: c801dd1e888e700699d5c47a50c1e454
Message 27: 375207,  Hash 27: a8ef9c52982bc618fd22920a01c0c903
Message 28: 625894,  Hash 28: edf759d85bd16a40d980bf1e0c62be88
Message 29: 603518,  Hash 29: 0b5451a36c86e420e00c625a25cfb8d6
Message 30: 787888,  Hash 30: e7d65f2cdb0ddc66b549e83107b71fa0
Message 31: 59456,  Hash 31: 7d55b99416ffdd7e5969d0afb035df2c
Message 32: 318410,  Hash 32: 4f161f8f56a4c671a87d1d2708748a99
Message 33: 30243,  Hash 33: 630571ee8e61fb9efaa9786a9de27353
Message 34: 658630,  Hash 34: 8d8aa93a63e676902347925a24897d96
Message 35: 292146,  Hash 35: 0b7ed02def9d32351f128bdd6800d58c
Message 36: 798720,  Hash 36: 18c058d29485860fdb522ee4914a85e7
Message 37: 919752,  Hash 37: c3fb58823aa5803f6283f357c22e4a47
Message 38: 637192,  Hash 38: cb78d80cbc291a8a4dc56ace2bab8119
Message 39: 870056,  Hash 39: f647265efb523e8e320430c6b947a7f0
Message 40: 955416,  Hash 40: 97bcacac874c82f9645c66b95ad4f9fe
Message 41: 189137,  Hash 41: 747f7d1ea81751cd298696aadfa1f946
Message 42: 183195,  Hash 42: a5a94f2b057aa4db7d0c42731a1b124b
Message 43: 916955,  Hash 43: d74b690cf647b737c4ec35a79adf6260
Message 44: 427124,  Hash 44: deb46012b8df52ab05a31b3d5544588c
Message 45: 524037,  Hash 45: 647396b9cc355eb98fb6ecbf4bb3a0d7
Message 46: 816854,  Hash 46: 20e299fd61533a98605102c73074732a
Message 47: 608116,  Hash 47: ae50f01859c5aa34c74a50afcec8c805
Message 48: 954331,  Hash 48: 63e43eaed9c13281622dd1d2bfaabc4d
Message 49: 368125,  Hash 49: 2541e7b58a76e4324c486141292a1600
```

سوال اضافه:

حمله lattice reduction به سیستم کوله پشتی یک تکنیک تحلیل رمزنگاری است که از نقاط ضعف در فرآیند رمزگذاری سیستم رمزنگاری کوله پشتی استفاده می کند. سیستم کوله پشتی از دنباله ای از اعداد صحیح به عنوان کلید عمومی استفاده می کند که از یک دنباله تصاعدی با استفاده از یک ضرب مخفی و محاسبات باقی مانده تولید می شود. فرایند رمزگذاری شامل تبدیل پیام متنی ساده به باینری و سپس ضرب هر بیت در بیت مربوطه در کلید عمومی و به دنبال ان جمع اوری محصولات است. حمله lattice reduction به سیستم کوله پشتی با ساختن یک lattice از کلید عمومی و سپس اعمال الگوریتم های lattice reduction برای حل مسئله جمع زیر مجموعه کار می کند. lattice با ایجاد یک ماتریس A که حاوی نمایش دودویی دنباله کلید عمومی است و سپس ساختن پایه ای برای lattice با استفاده از ستون های A ساخته می شود سپس الگوریتم‌های lattice reduction روی پایه اعمال می‌شوند تا پایه کوتاه تر و متعامد تری پیدا کنند که بتوانند از ان برای حل مسئله جمع زیر مجموعه استفاده کرد. الگوریتم Lenstra-Lenstra-Lovasz یا LLL یکی از این الگوریتم هایی است که می توان برای این منظور استفاده کرد. هنگامی که lattice کاهش یافت، پایه ای پیدا می شود که کوتاه تر از پایه اصلی است. از این مبنا می توان برای حل مسئله جمع زیر مجموعه با یافتن یک نقطه lattice نزدیک به مجموع هدف استفاده کرد سپس نقطه lattice را می توان به یک راه حل برای مسئله جمع زیر مجموعه تبدیل کرد که می تواند برای بازیابی ضریب مخفی و رمزگشایی پیام استفاده شود.

برای محافظت در برابر حملات lattice reduction در سیستم کوله پشتی، می توان از اندازه های بزرگتر کلید استفاده کرد و کلید عمومی را می توان با استفاده از الگوریتم پیچیده تری که در برابر حملات lattice مقاوم است تولید کرد. یکی از این الگوریتم ها سیستم رمزنگاری کوله پشتی Chor-Rivest است که از یک جایگشت trapdoor برای تولید کلید عمومی استفاده می کند.