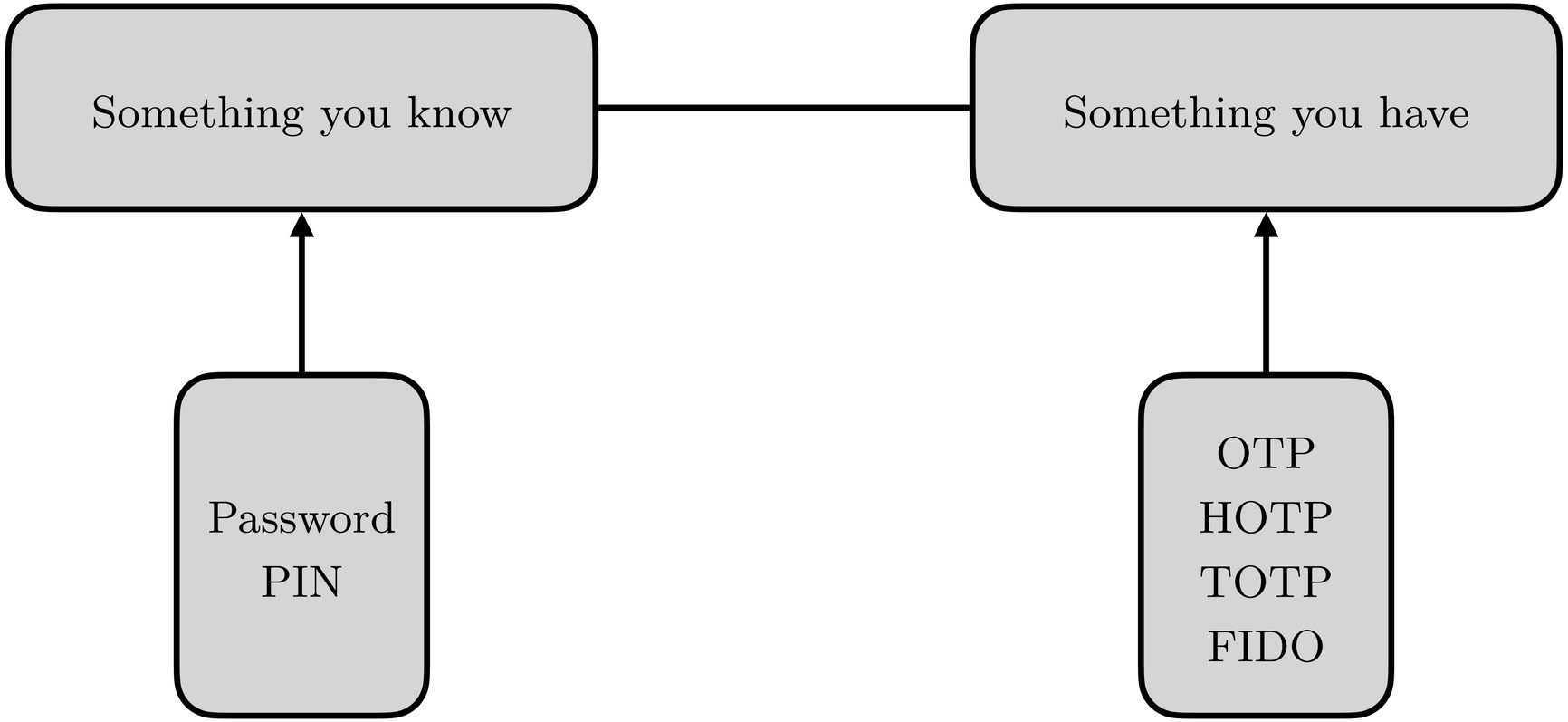
Multi-factor Authentication

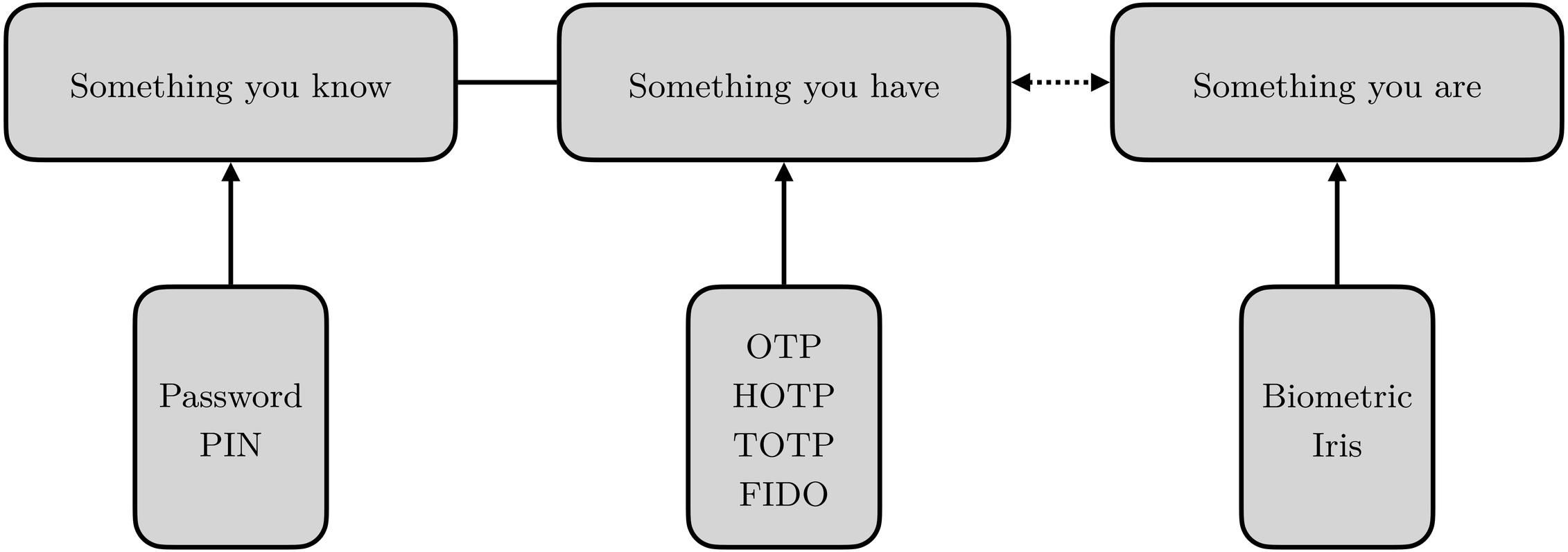
(A crash-course)

Autumn Luzovich (*they/them*) — 2024

How does it work?


Something you know
Something you have
Password
PIN
OTP
HOTP
TOTP
FIDO

2FA


Something you know
Something you have
Something you are
Password
PIN
OTP
HOTP
TOTP
FIDO
Biometric
Iris

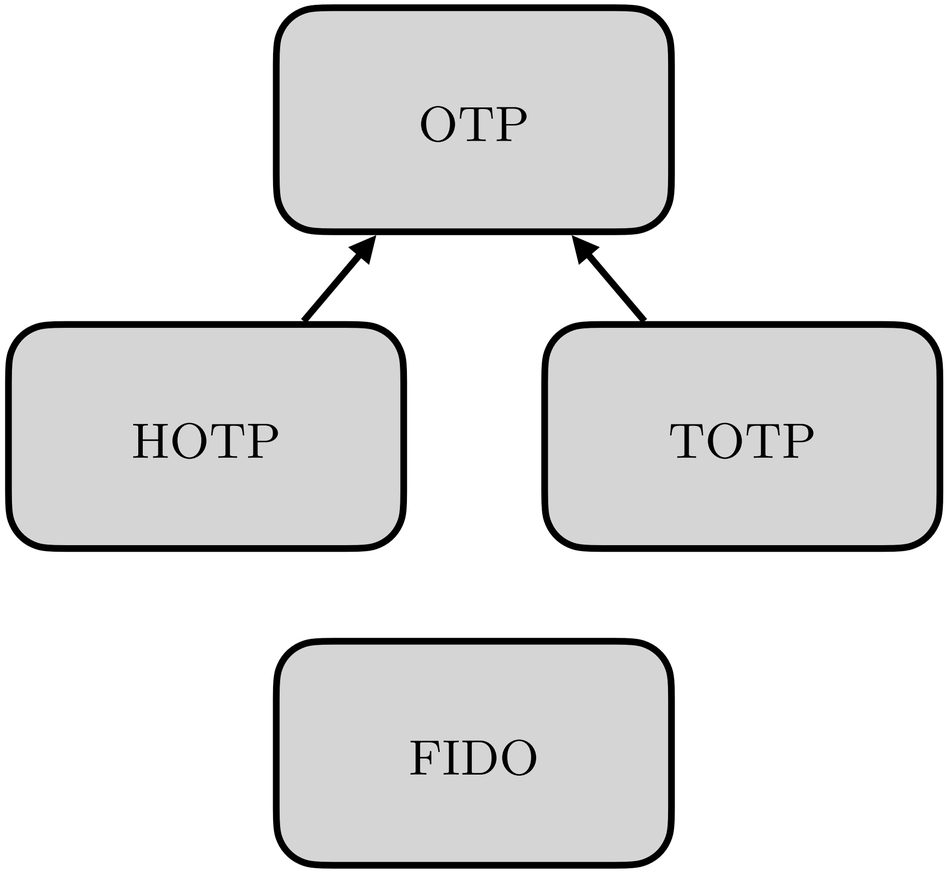
MFA

Something you know

- Password
- PIN (Personal Identification Number)

Something you have

- OTP (One-Time Password)
  - HOTP (Hash-based OTP)
  - TOTP (Time-based OTP)
- FIDO (Fast IDentity Online)
  - U2F (Universal 2nd Factor)
  - CTAP2 – (Client to Authenticator Protocol)



Something you have

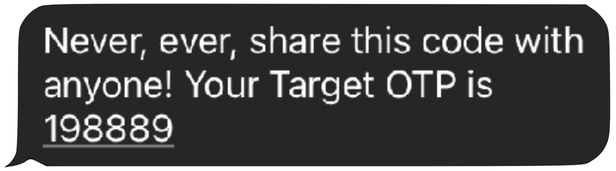
One-Time Password

- Typically sent over SMS/Email/Voicemail
- Also umbrella term for HOTPs and TOTPs

Something you have

One-Time Password

- Typically sent over SMS/Email/Voicemail
- Also umbrella term for HOTPs and TOTPs



Something you have

One-Time Password

- Typically sent over SMS/Email/Voicemail
- Also umbrella term for HOTPs and TOTPs

Step Two App, https://neilsardesai.com/step-two

One-Time Password

HOTPs & TOTPs

- Use 3–4 variables in calculation
- Differ in how one of those variables are calculated

TOTP	HOTP
Digit count (min of 6 is standard)	
Shared key	
Time contingent	Counter contingent

One-Time Password

HOTPs & TOTPs

- Use 3–4 variables in calculation
- Differ in how one of those variables are calculated

TOTP	HOTP
Length(6) → “293 842” Length(8) → “0148 1928”	
Shared key	
Time contingent	Counter contingent

One-Time Password

HOTPs & TOTPs

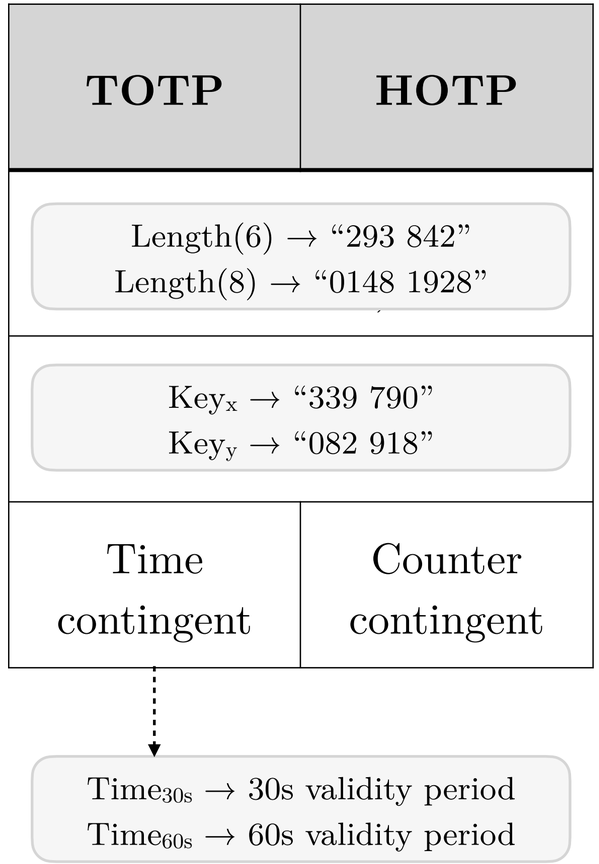
- Use 3–4 variables in calculation
- Differ in how one of those variables are calculated

TOTP	HOTP
Length(6) → “293 842” Length(8) → “0148 1928”	
Key_x → “339 790” Key_y → “082 918”	
Time contingent	Counter contingent

One-Time Password

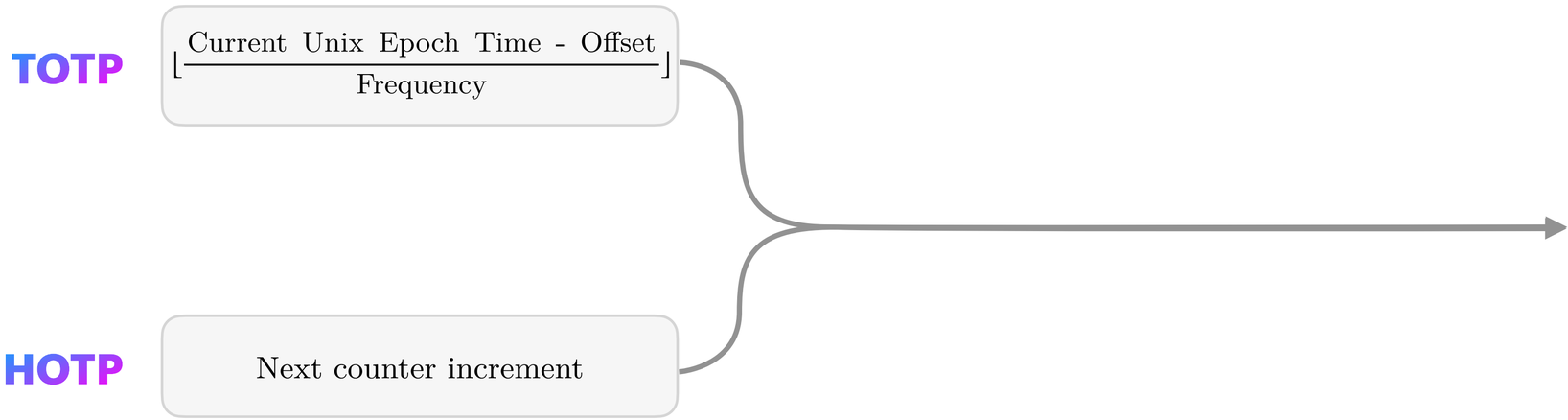
HOTPs & TOTPs

- Use 3–4 variables in calculation
- Differ in how one of those variables are calculated



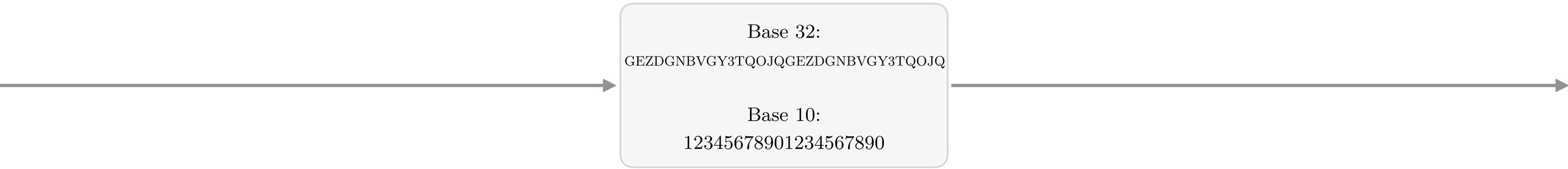
HOTPs & TOTPs

Calculation



Step 1 — Getting the Counter

HOTPs & TOTPs
Calculation



Step 2 — Decode secret from Base 32 (if needed)

HOTPs & TOTPs
Calculation

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	0	3	2	5	4	7	6	9	8	B	A	D	C	F	E
2	2	3	0	1	6	7	4	5	A	B	8	9	E	F	C	D
3	3	2	1	0	7	6	5	4	B	A	9	8	F	E	D	C
4	4	5	6	7	0	1	2	3	C	D	E	F	8	9	A	B
5	5	4	7	6	1	0	3	2	D	C	F	E	9	8	B	A
6	6	7	4	5	2	3	0	1	E	F	C	D	A	B	8	9
7	7	6	5	4	3	2	1	0	F	E	D	C	B	A	9	8
8	8	9	A	B	C	D	E	F	0	1	2	3	4	5	6	7
9	9	8	B	A	D	C	F	E	1	0	3	2	5	4	7	6
A	A	B	8	9	E	F	C	D	2	3	0	1	6	7	4	5
B	B	A	9	8	F	E	D	C	3	2	1	0	7	6	5	4
C	C	D	E	F	8	9	A	B	4	5	6	7	0	1	2	3
D	D	C	F	E	9	8	B	A	5	4	7	6	1	0	3	2
E	E	F	C	D	A	B	8	9	6	7	4	5	2	3	0	1
F	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0

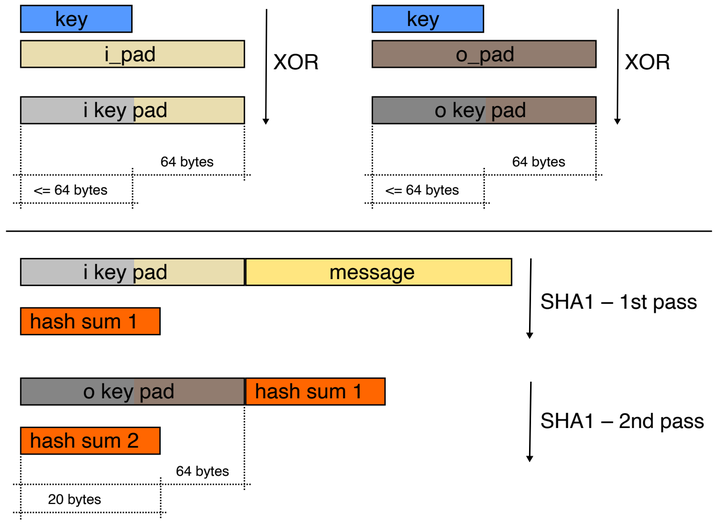
XOR Table for Hex, https://crypto.stackexchange.com/questions/43200/how-to-xor-two-hexa-numbers-by-hand-fast

$$\mathrm{HMAC}(K, m) = \mathrm{H}\Big(\big(K' \oplus opad\big) \parallel \mathrm{H}\big(\big(K' \oplus ipad\big) \parallel m\big)\Big)$$

$$K' = \begin{cases} \mathrm{H}(K) & \text{if } K \text{ is larger than block size} \\ K & \text{otherwise} \end{cases}$$

Step 3 — Calculate SHA1 HMAC (Hash-based Message Authentication Code)

HOTPs & TOTPs
Calculation



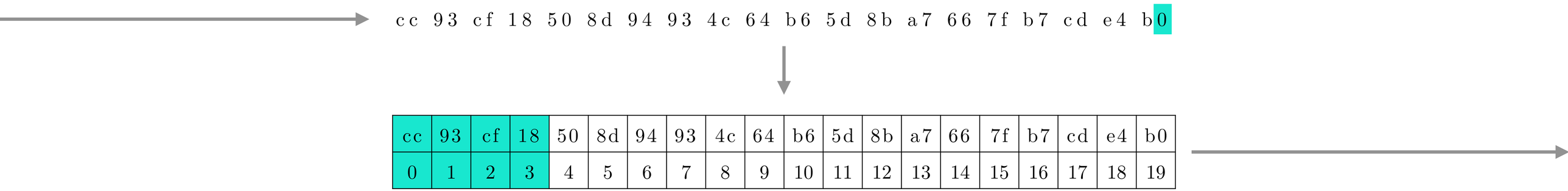
By Gdrooid - Own work, CC0, https://commons.wikimedia.org/w/index.php?curid=34446189

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	0	3	2	5	4	7	6	9	8	B	A	D	C	F	E
2	2	3	0	1	6	7	4	5	A	B	8	9	E	F	C	D
3	3	2	1	0	7	6	5	4	B	A	9	8	F	E	D	C
4	4	5	6	7	0	1	2	3	C	D	E	F	8	9	A	B
5	5	4	7	6	1	0	3	2	D	C	F	E	9	8	B	A
6	6	7	4	5	2	3	0	1	E	F	C	D	A	B	8	9
7	7	6	5	4	3	2	1	0	F	E	D	C	B	A	9	8
8	8	9	A	B	C	D	E	F	0	1	2	3	4	5	6	7
9	9	8	B	A	D	C	F	E	1	0	3	2	5	4	7	6
A	A	B	8	9	E	F	C	D	2	3	0	1	6	7	4	5
B	B	A	9	8	F	E	D	C	3	2	1	0	7	6	5	4
C	C	D	E	F	8	9	A	B	4	5	6	7	0	1	2	3
D	D	C	F	E	9	8	B	A	5	4	7	6	1	0	3	2
E	E	F	C	D	A	B	8	9	6	7	4	5	2	3	0	1
F	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0

XOR Table for Hex, https://crypto.stackexchange.com/questions/43200/how-to-xor-two-hexa-numbers-by-hand-fast

Step 3 — Calculate SHA1 HMAC (Hash-based Message Authentication Code)

HOTPs & TOTPs
Calculation



Step 4 — Dynamically truncate result using last byte

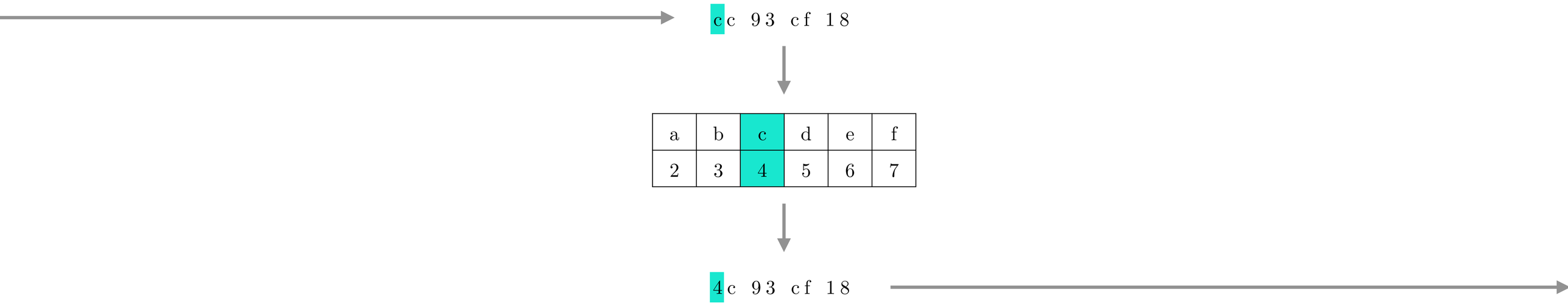
HOTPs & TOTPs
Calculation

cc 93 cf 18 50 8d 94 93 4c 64 b6 5d 8b a7 66 7f b7 cd e4 bf

cc	93	cf	18	50	8d	94	93	4c	64	b6	5d	8b	a7	66	7f	b7	cd	e4	bf
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

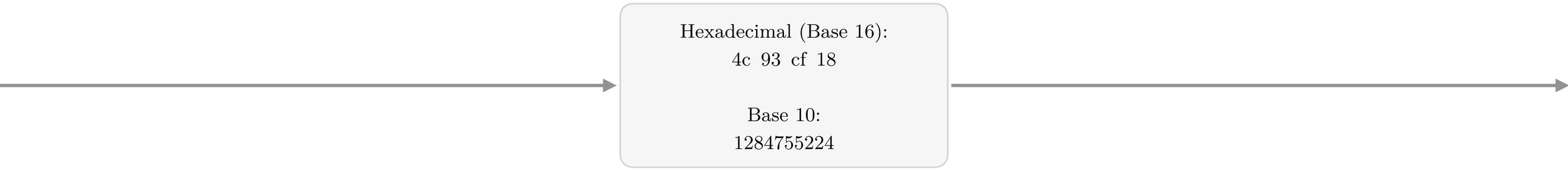
Step 4 — Dynamically truncate result using last byte

HOTPs & TOTPs
Calculation



Step 5 — Clear top of selection (if necessary)

HOTPs & TOTPs
Calculation



Step 6 — Convert to Base 10

HOTPs & TOTPs
Calculation

Code	1	2	8	4	7	5	5	2	2	4
Length	10	9	8	7	6	5	4	3	2	1

We’re done!

Step 7 — Grab selection by code length

Something you have

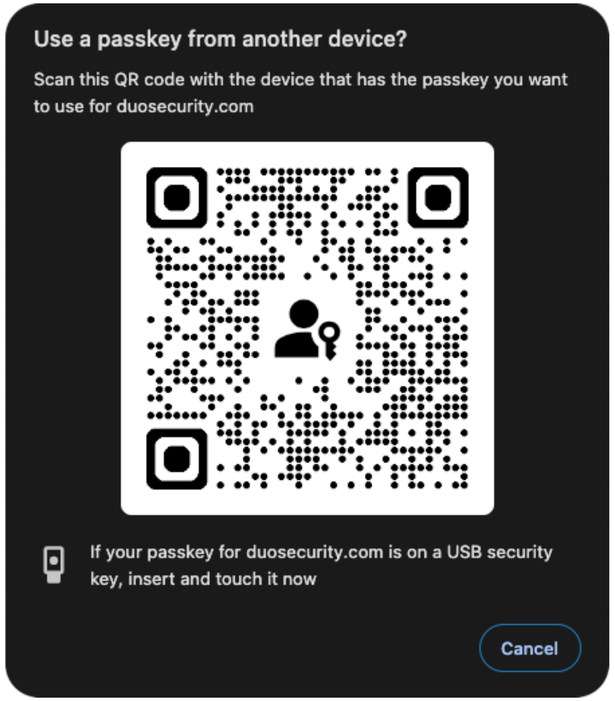
FIDO — CTAP1 / U2F

- Primarily seen in “security keys”
- Only two major flows: Registration & Authentication
- Highly resistant to phishing because of ID matching
- Stems into FIDO2; CTAP2; WebAuthn; “Passkeys”

Something you have

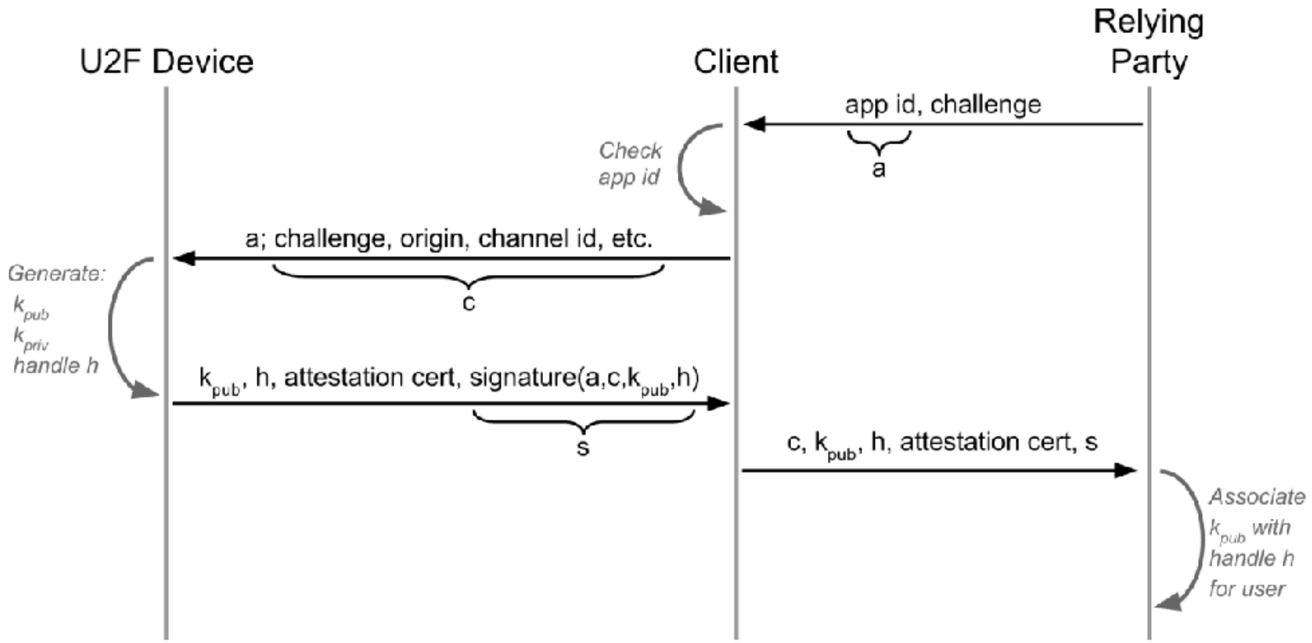
FIDO — CTAP1 / U2F

- Primarily seen in “security keys”
- Only two major flows: Registration & Authentication
- Highly resistant to phishing because of ID matching
- Stems into FIDO2; CTAP2; WebAuthn; “Passkeys”



FIDO — CTAP1 / U2F

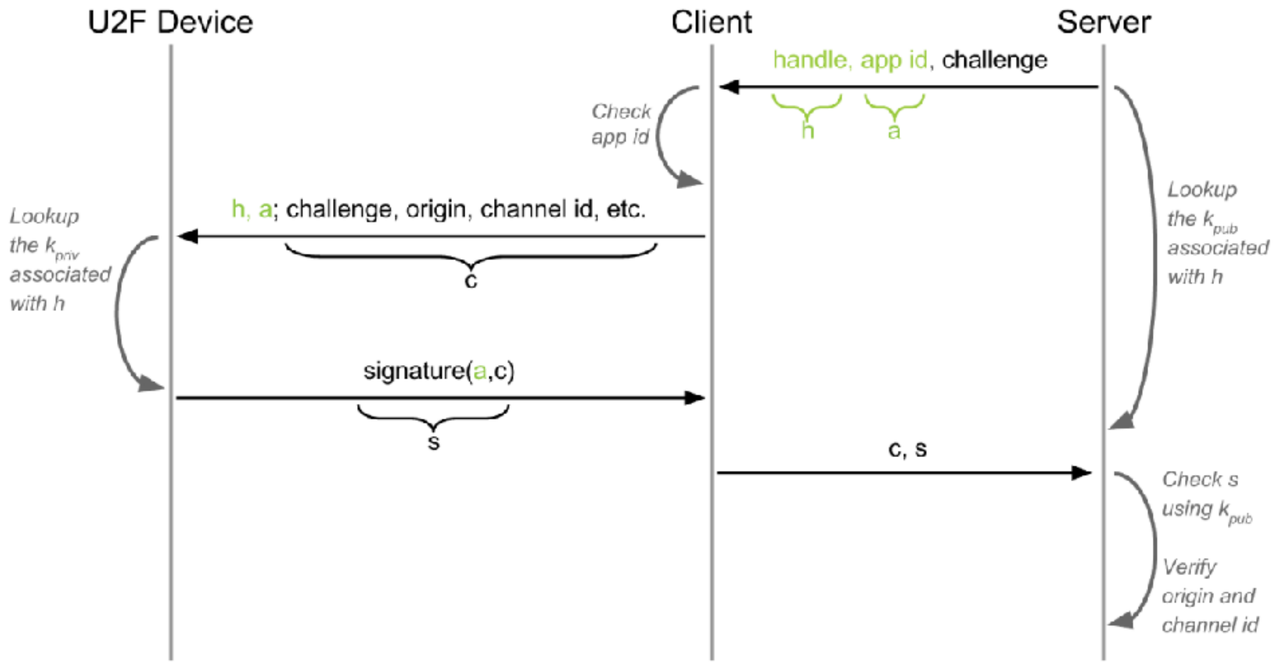
Registration



CTAP1/U2F Registration Flow, https://engineering.tumblr.com/post/145560228370/u2f-with-yubikeys

FIDO — CTAP1 / U2F

Authentication



CTAP1/U2F Authentication Flow, https://engineering.tumblr.com/post/145560228370/u2f-with-yubikeys

Security Considerations

- OTP
- HOTP
- TOTP
- FIDO

References

OTP, TOTP, HOTP

- https://mikecat.github.io/sbs_totp/
- https://jacob.jkrall.net/totp
- RFC 6238 — TOTP
- RFC 4648 — Base16, Base32, and Base64 Encodings
- RFC 4225 — HOTP
- RFC 2104 — HMAC

FIDO U2F/CTAP1

- https://docs.yubico.com/yesdk/users-manual/application-u2f/how-u2f-works.html
- https://fidoalliance.org/specs/fido-v2.0-ps-20190130/fido-client-to-authenticator-protocol-v2.0-ps-20190130.html
- https://webauthn.io/
- https://webauthn.guide/
- https://webauthn.me/