Analyzing Compliance and Complications of Integrating Internationalized X.509 Certificates

Mingming Zhang[1] and Jinfeng Guo[2], Yiming Zhang[3], Shenglin Zhang[2], Baojun Liu[3], Hanqing Zhao[3], Xiang Li[2], Haixin Duan[3,4]

[1]Zhongguancun Laboratory [2]Nankai University [3]Tsinghua University [4]Quancheng Laboratory

Presenter: Mingming Zhang
Zhongguancun Laboratory

2026.03

PKI and X.509 Certificates

- **Public Key Infrastructure (PKI)** is the security foundation of the Internet.
- **X.509 Certificates:**

  Binding identities (e.g., hostnames) to cryptographic keys.

Certificate Authority (CA)

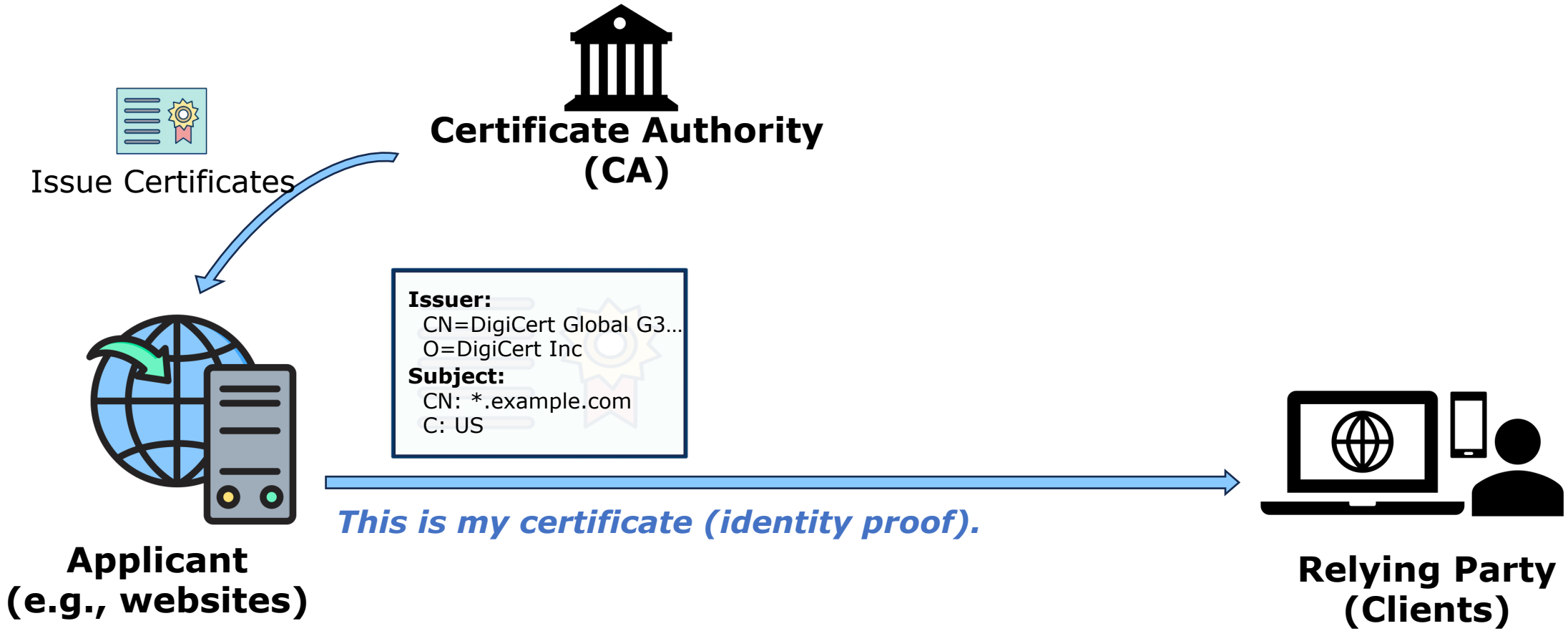
Applicant (e.g., websites)

Relying Party (Clients)

PKI and X.509 Certificates

- **Public Key Infrastructure (PKI)** is the security foundation of the Internet.
- **X.509 Certificates:**

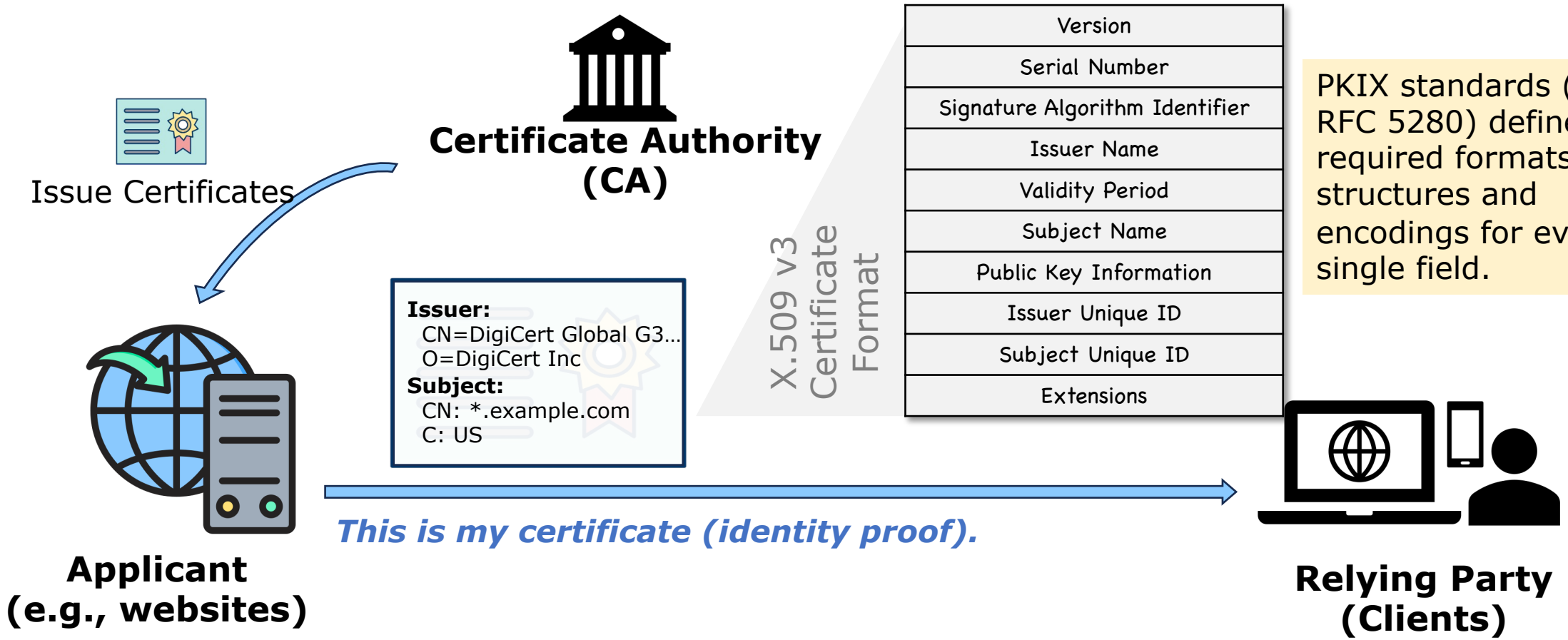
  Binding identities (e.g., hostnames) to cryptographic keys.



PKI and X.509 Certificates

- **Public Key Infrastructure (PKI)** is the security foundation of the Internet.
- **X.509 Certificates:**

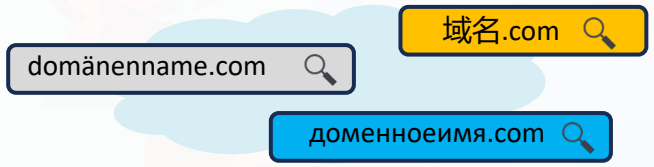
  Binding identities (e.g., hostnames) to cryptographic keys.



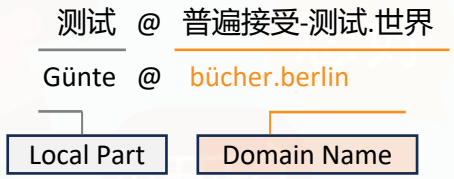
PKIX standards (e.g., RFC 5280) define the required formats, structures and encodings for every single field.

Unicerts: X.509 Certificates with Internationalized Contents

- **Universal Acceptance (UA) :** An initiative for a truly multilingual and digitally inclusive Internet, ensuring Internet applications/systems (e.g., PKI) can properly handle characters beyond the printable ASCII set.

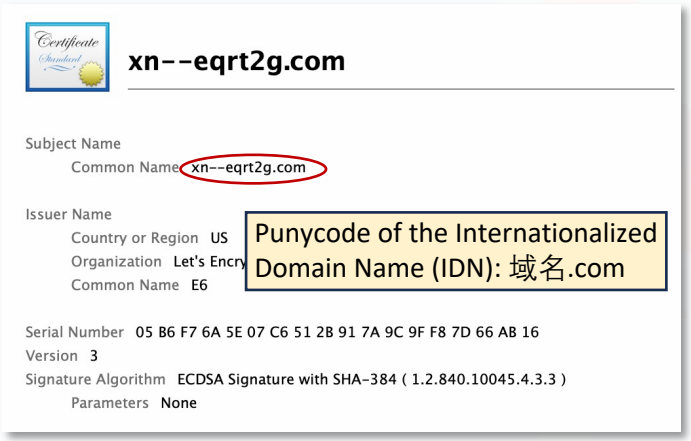


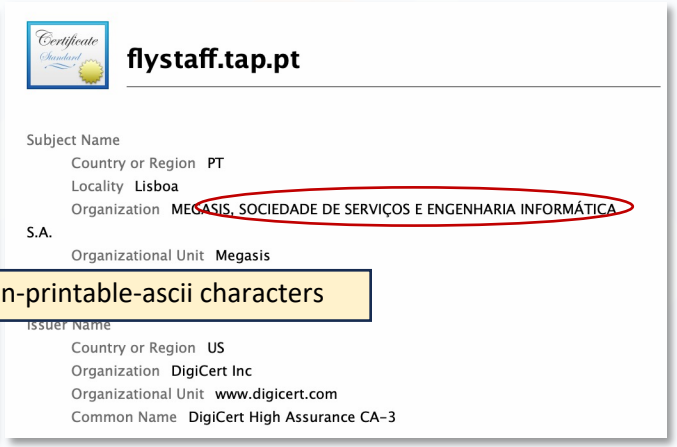
Internationalized Domain Name



Internationalized Email

- **(Uni)certs:** X.509 certificates containing internationalized contents, e.g., domain names (IDNs), resource identifiers (IRIs), or multilingual text with non-printable-ASCII set.





Contain non-printable-ascii characters

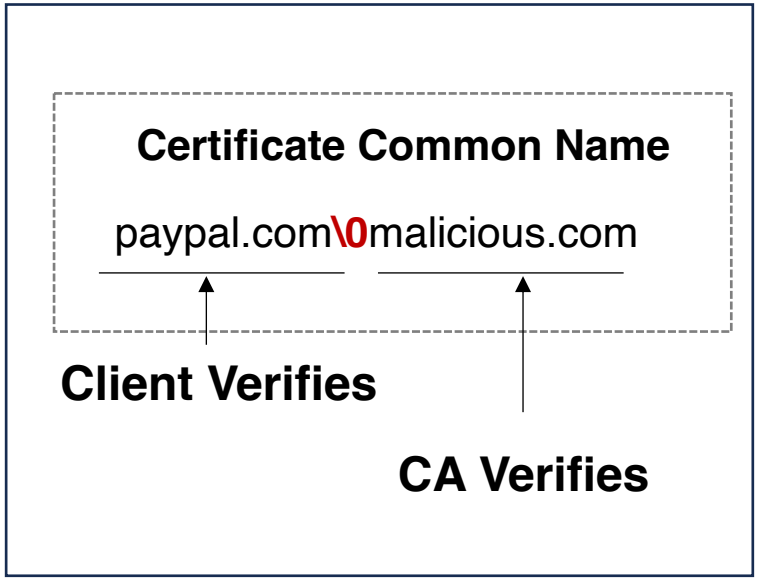
Motivation: Why Unicerts Break Things

- The broader Unicode space complicates the issuance, parsing, and validation of X.509 certificates, leading to security or usability issues.

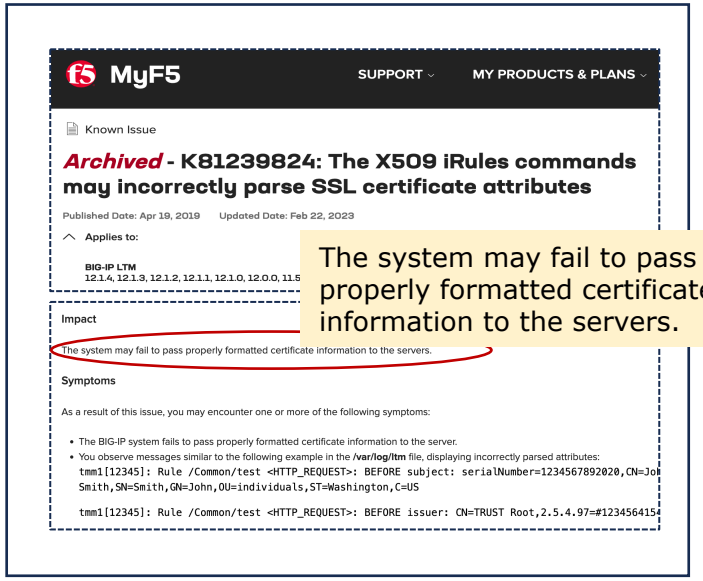
Motivation: Why Unicerts Break Things

- The broader Unicode space complicates the issuance, parsing, and validation of X.509 certificates, leading to security or usability issues.
- **Real-World incidents:**

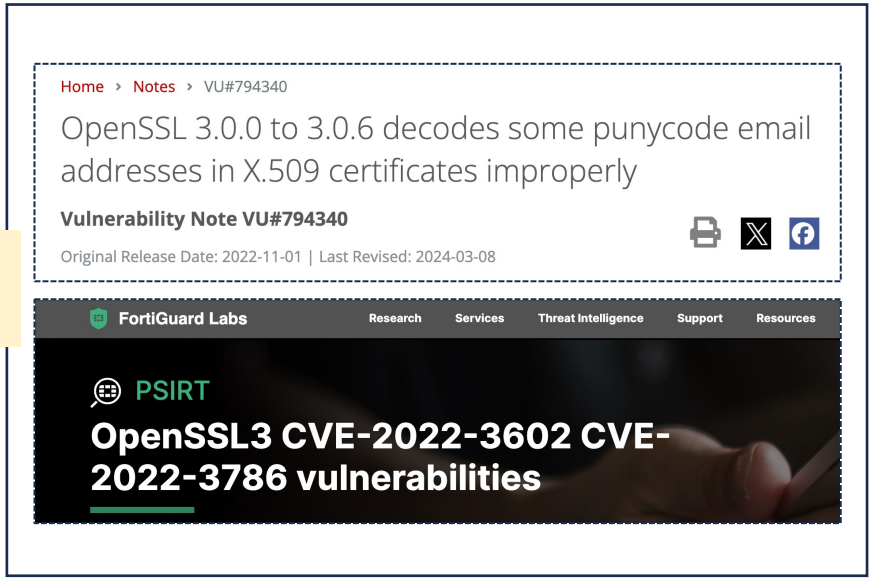
  Improper Unicode handling in certificates can cause:



Certificate spoofing



Incorrect attribute parsing

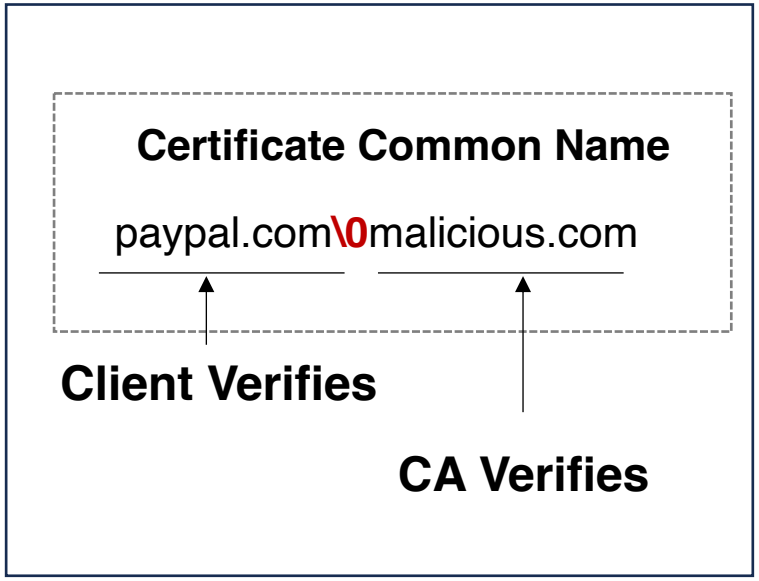


Buffer overflows

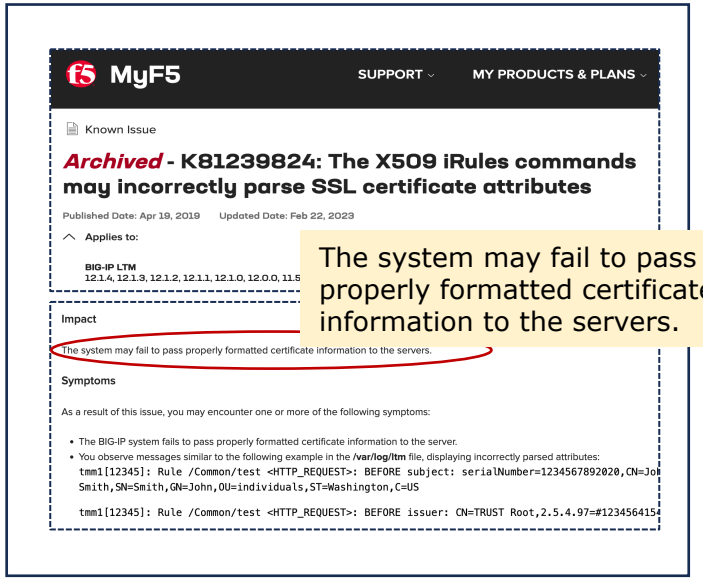
Motivation: Why Unicerts Break Things

- The broader Unicode space complicates the issuance, parsing, and validation of X.509 certificates, leading to security or usability issues.
- **Real-World incidents:**

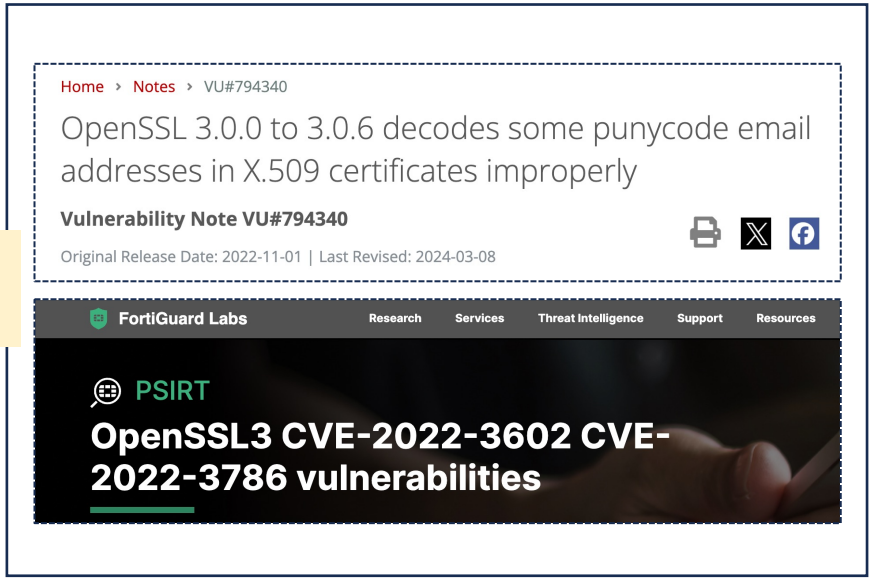
  Improper Unicode handling in certificates can cause:



Certificate spoofing



Incorrect attribute parsing



Buffer overflows

The gap: While X.509 standards support Unicode, the UA readiness of PKI's core mechanisms (issuance, parsing) remains **largely untested.**

Our Research Questions

- We conducted the first large-scale measurement and empirical study of Unicerts

RQ1 Issuance Compliance

Have CAs issued Unicerts in compliance with the complex Unicode-related standards?

RQ2 Parsing Accuracy

Do mainstream TLS implementations correctly parse Unicerts according to normative constraints?

RQ3 Real-World Impact

What are the security and usability impacts of noncompliant issuance and parsing flaws?

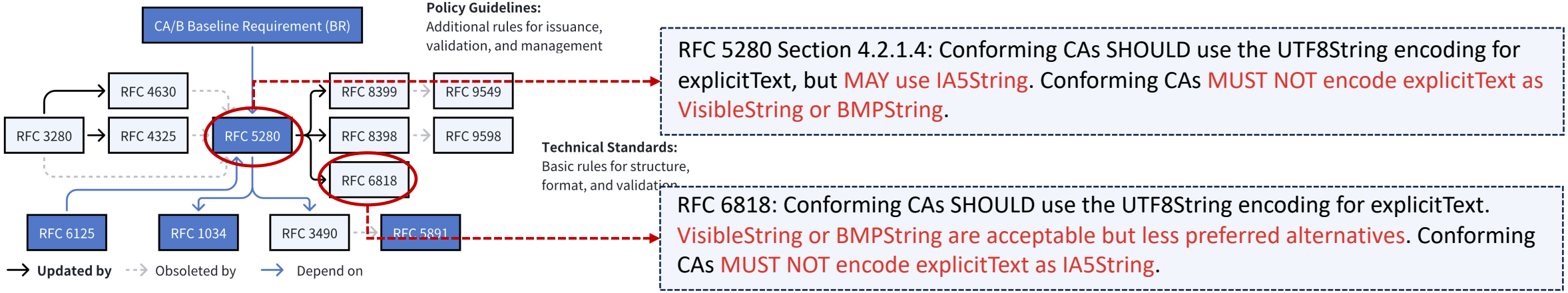
RQ1: Issuance Compliance of Unicerts

Identify a definitive standard: how can we establish clear normative requirements for Unicert compliance?

➢ Mastering the specifications

❑ Challenges

1. Relevant specifications (e.g., X.509 & updates, DNS & IDN specs, CA/B BRs) are highly **interdependent and evolving**.
2. Many rules for formats, structure, and encodings of X.509 certificate fields are scattered across natural language, ASN.1 definitions, and tables, etc.



E.g., Requirements evolve through periodic updates and revisions.

RQ1: Issuance Compliance of Unicerts

- **Mastering the specifications.**
  - **Our solution:**
    - Employing RFCGPT to navigate complex requirements and constraints.

      PKIX specs (RFC 5280), updates (RFC 9549, 9598, 6818), references (RFC 3490, 1034, 3454), dependent standards (RFC 6125, IDNA suites), and CA/B BRs.

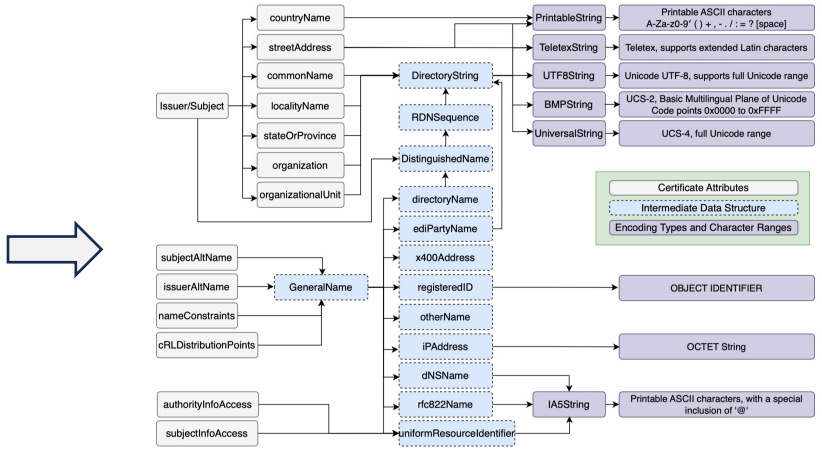
    - Prompt example: identifying encoding, structure, and character constraints for cert fields that allow non-ascii characters.

Prompt Template

Instruction:
Present relationships among the following data structures and encoding types: [DistinguishedName, RDNSequence, UTF8String...]. Use '-->' to denote a relationship between them.
Requirements:
1. If a structure contains others, describe each layer in sequence.
2. Point each ASN.1 structure directly to its encoding types.
3. Point each encoding type to the character sets.

Example:
If a structure GeneralName contains DNSName, and DNSName should be serialized as IA5String, your output would be:
GeneralName-->DNSName-->IA5String (ASCII characters, excluding '@')

Desired Output Format:
Shot-1: DistinguishedName-->RDNSequence-->DirectoryString
Shot-2: GeneralName-->IPAddress (binary/octet string)

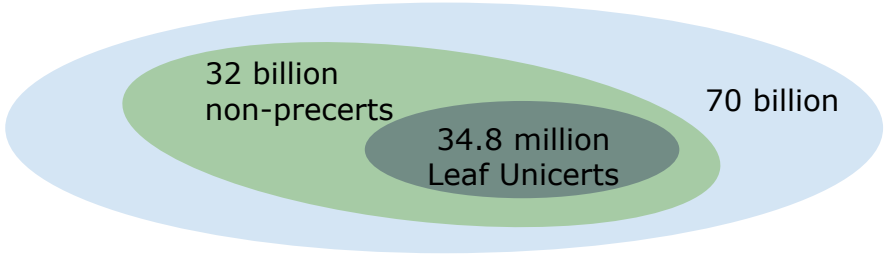


Results: Extracted 95 rules for 36 Unicode-related fields, with 50 missing from existing compliance checks.

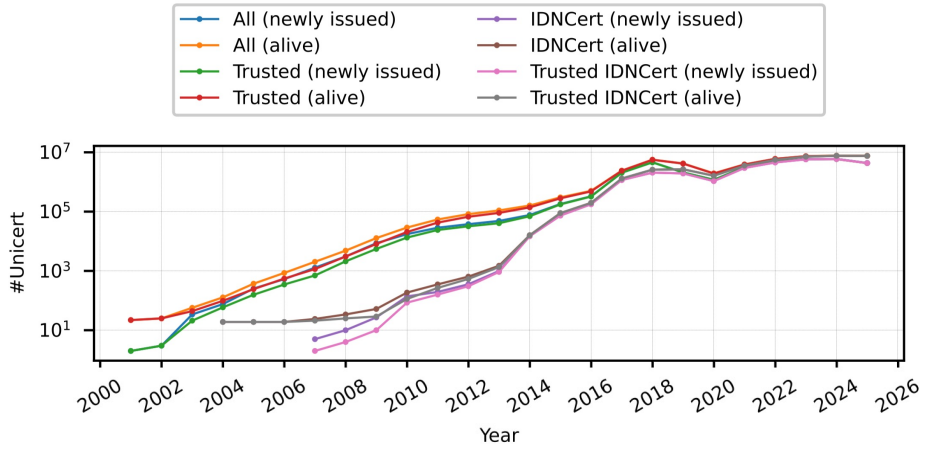
Overview of Unicert Issuance

➢ Measurement

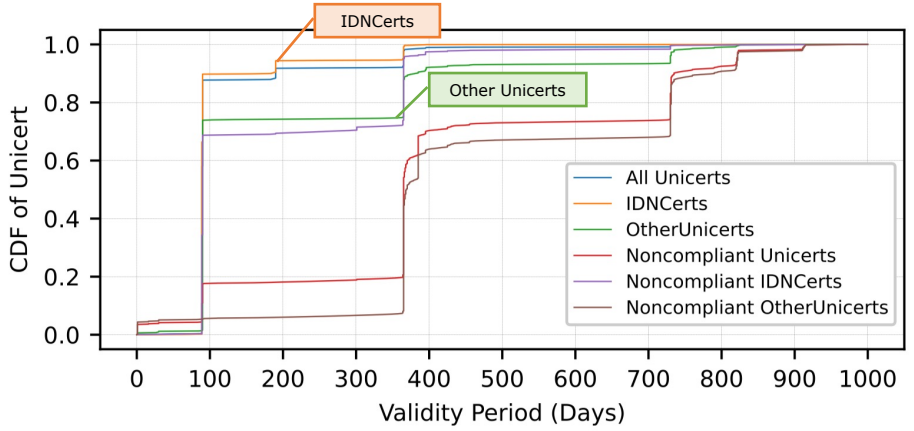
- ❑ Checking **34.8 Million Unicerts** from a 70 Billion CT certificate dataset, against the 95 rules.



- ❑ **Unicert issuance ecosystem:** These Unicerts were issued by **4,528** CA certificates across **698** issuer organizations (97.6% from the trusted CAs).



Unicert issuance numbers are clearly rising.



Unicerts containing only IDNs have shorter validity periods.

Quantifying Real-world Noncompliance

- **We identified 249K noncompliant Unicerts from the CT dataset.**
  - 65.3% issued by publicly trusted CAs; 21.1% from issuers with limited trust; 13.6% from untrusted issuers
- **Noncompliance issue categories**
  - T1. Improper character checks (43K Unicerts):
    i. Malformed strings: e.g., non-printable characters in PrintableString
    ii. Disallowed characters: e.g., control characters in UTF8String
  - T2. Lack of value normalization (3 certs):
    i. ExplicitText (UTF8String) not normalized to Unicode Normalization Form C (NFC)
    ii. IDNs not normalized to NFC after converting Punycode (xn-- string) to Unicode
  - T3. Invalid format/structure (206K certs):
    i. Illegal format: basic formatting errors, which can hinder parsing
    ii. Invalid encoding: e.g., encoding CN with TeletexString/BMPString instead of IA5String
    iii. Invalid structure: violations of structural rules that hinder parsing
    iv. Discouraged field: e.g., CN in Subject or URI in SAN

Quantifying Real-world Noncompliance

- Noncompliant Unicerts were issued by 505 issuer organizations, covering 78 CA owners in CCADB and 295 untrusted/unknown issuers.

Table: Top 10 issuer organization names by noncompliant unicerts.

Issuer OrgName	Trust status	Region	Noncompliant Unicerts	Recently Issued
Česká pošta...	◐	CZ	22,939 (96.39%)	0
Symantec Corporation	○	US	18,092 (51.47%)	0
Dreamcommerce S.A.	●	PL	17,291 (44.83%)	0
DigiCert Inc	●	US	17,276 (3.40%)	40
Let's Encrypt	●	US	15,484 (0.06%)	7,091
StartCom Ltd.	○	IL	14,168 (72.97%)	0
COMODO CA Limited	◐	GB	11,870 (0.25%)	0
ZeroSSL	●	AT	11,224 (2.53%)	4,094
Government of Korea	◐	KR	10,416 (87.33%)	0
VeriSign, Inc.	●	US	7,513 (59.12%)	0
Other	-	-	103,008 (0.29%)	1,802
Total	-	-	249,281 (0.72%)	13,027

● publicly trusted ○ untrusted ◐ trusted in specific regions/scenarios

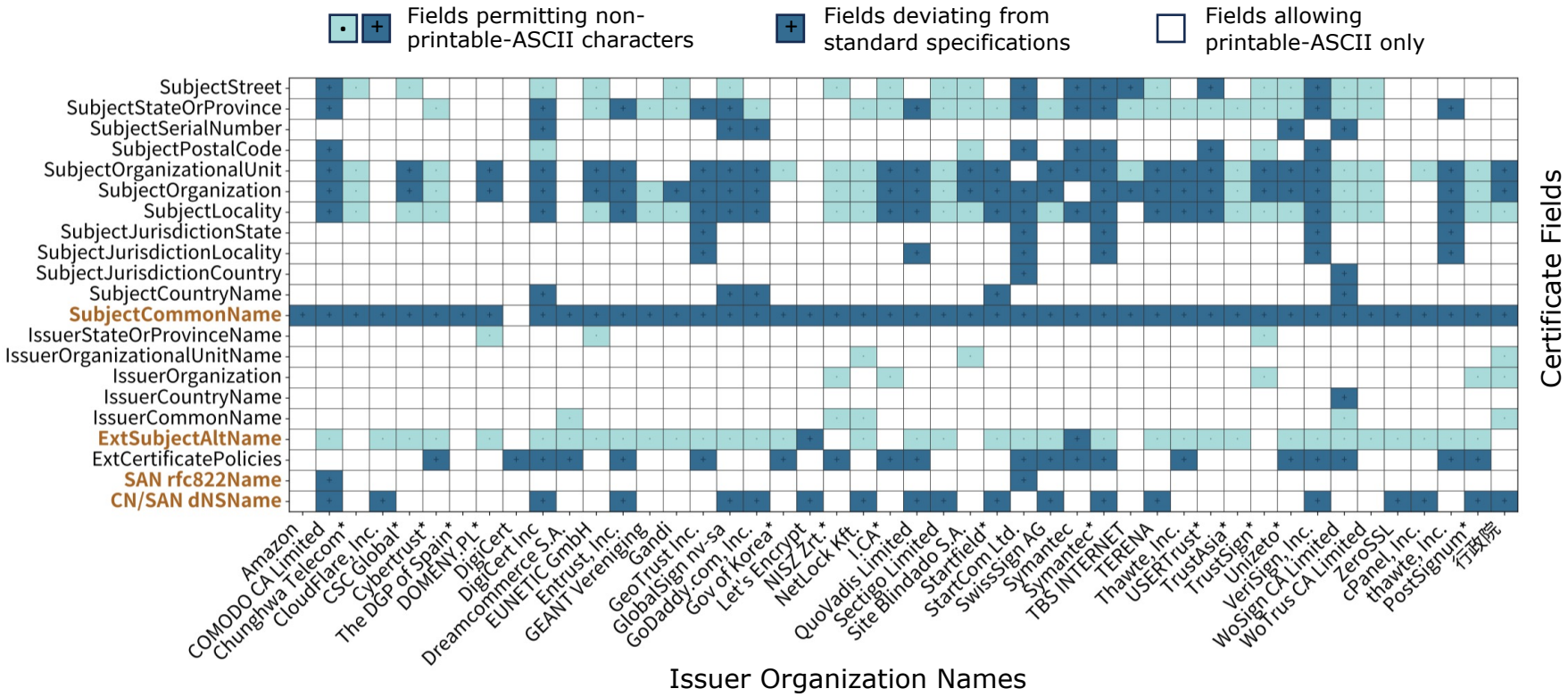
Finding 1: The problematic issuances involve both major global CAs and regional providers.

Finding 2: Some internationalization requirements may not be fully covered by the linting tools CAs currently use.

Finding 3: Automated domain validation workflows limiting field customization (e.g., Let's Encrypt allowing only DNSNames) may help reduce noncompliance.

Troublesome Certificate Fields

- Issuers show **inconsistent** and **noncompliant** handling of certain non-ASCII fields.
- The encoding types, character ranges, or formats of **17 subject and extension fields** were deviated from standard or baseline requirements.



21 fields permitting non-printable-ASCII chars or IDNs/IEAs/IRIs

Case Study: The Problems in Certificate Fields

Poor validation of DNSNames

DNSNames in certs are critical for identifying peer entities.

- We found 27K Unicerts with malformed IDNs:
  - i. cannot convert to Unicode
  - ii. include illegal characters (e.g., bidirectional controls) after Punycode decoding
- Example: DNSNames in the SubjectAltNames extensions

Punycode IDN (A-Label, ASCII Compatible)

xn--2ug.walesbonner.net
xn--strandkrbe-kcapi.aosz.dev.bigfish.hu
blog.xn--vpr-xda.statgrad.org

⟺

Unicode (U-Label)

<0x200e>.walesbonner.net
str<0x89><0x87>and<0x88>krbe.aosz.dev.bigfish.hu
blog.<0xa0>vpr.statgrad.org

Syntactically allowed under the current CA/B Baseline Requirements!

✓ Valid P-Labels
✓ Resolvable via wildcard DNS: allowing domain validation
✗ Valid IDNs: U-Labels contain **DISALLOWED** characters

Deviation from multiple related standards?

RFC 5280: Implementations MUST convert IDNs to ASCII Compatible Encoding (ACE) per **RFC3490** before storage in the dNSName. → **IDNA2003 suites** (RFC 3490 →RFC 3491→RFC 3454

CA/B BR: P-Labels not conforming to IDNA 2003 are allowed to support newer Unicode versions and updated IDNA standards. → **IDNA2008 suites (**RFC 5890 →RFC 5891→RFC5892)

→ U+200E, U+0080-U+009F, and U+00A0 are DISALLOWED characters

CA side: Compliant

User agent side: Problematic

RQ2: Parsing Flaws

Do TLS libraries respect declared encodings and enforce strict character checks?

- **Methodology: Gray-box testing for TLS implementations.**
  - **Goal**
    - Check whether TLS implementations properly decode and validate Unicert character ranges per normative requirements.
  - **Test cases: Constructed special test Unicerts using diverse:**
    - Unicode blocks (e.g., C0 Controls)
    - Encoding types (e.g., PrintableString, UTF8String, BMPString).
  - **Test scope**
    - The certificate parsing APIs in 9 mainstream TLS libraries.

      OpenSSL, GnuTLS, Forge, PyOpenSSL, Cryptography, Golang Crypto, BouncyCastle, Java.security.cert, Node.js Crypto

Decoding and Character Handling Anomalies are Common

- We uncovered certificate field decoding or character handling anomalies in **ALL 9 tested TLS libraries.**
- Decoding Issues (Inconsistent Identity):
  i. **Incompatible Decoding:** Decoding values using non-standard-declared methods.
  e.g., Forge decodes UTF8String using ISO-8859-1.
  ii. **Over-Tolerant Decoding:** Decoding values using methods that permit a broader character range than defined by the standard encoding.
  e.g., GnuTLS decodes PrintableString using UTF-8.
  iii. **Modified Decoding:** Replacing undecodable bytes with substitute characters instead of rejecting them.
- Character Handling Issues (Validation Bypass):
  i. Every library exhibited standard violations in handling special characters (e.g., improper escaping, accepting characters beyond standard ranges).

Case Study: Potential Security Implications

Case I. Encoding-Decoding mismatches could allow Common Name forgery.

Encoding by CA: BMPString

\u7777\u772E\u6F72\u6163\u6C65\u2E63\u6F6D
(CJK Unified Ideographs:睷眲淵慣伎　潭)

Decoding by library: ASCII

www.oracle.com

Case II. Improper replacement of control character could enable CRL spoofing

URI in CRLDistributionPoints: (UTF8String)

http://ssl\u0001test.com

After replacement

http://ssl.test.com

Case III. Allowing non-DNS characters in DNSNames can enable attribute embedding

Field value of a DNSName: DNSName="a.com DNS:b.com"

X.509-text representation: "DNS:a.com DNS:b.com"

String-based analyzers can misinterpret it as valid for two domain names.

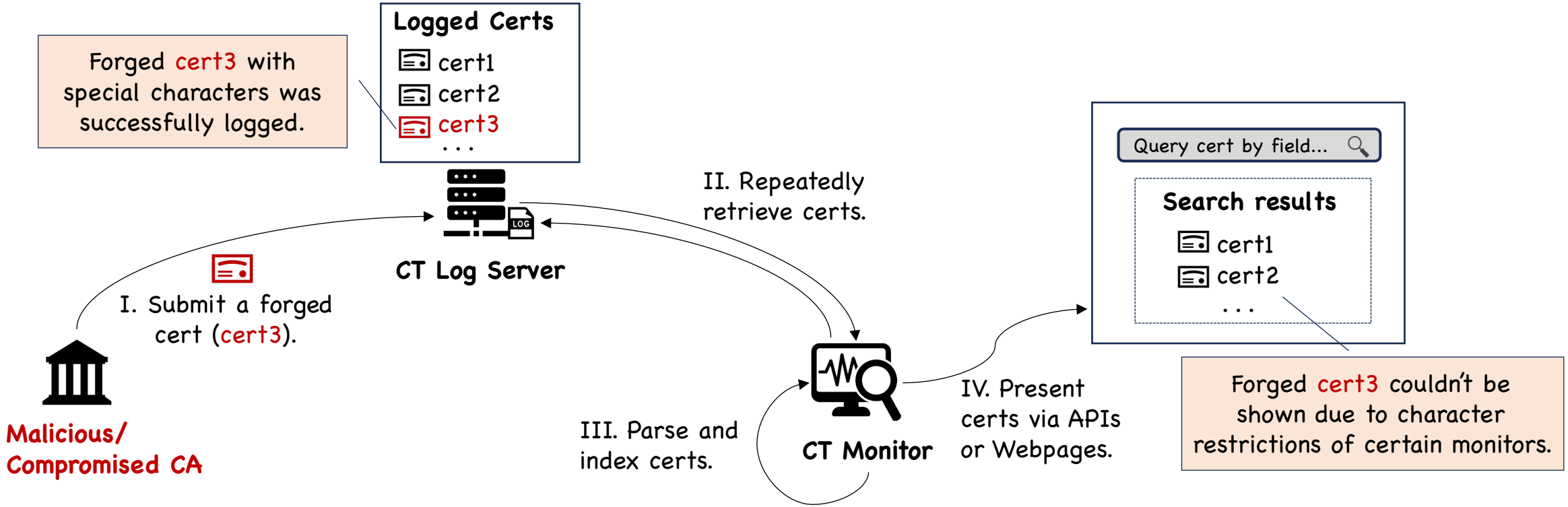
Takeaways:

1. These threats are uncommon, as they require both flawed issuers and parsers.
2. We did found certificates in CT logs that could be misinterpreted, but no evidence of real exploitation.
3. Ongoing PKI practices may help reduce some risks:
   e.g., discouraging CN-based validation, using short-lived certificates to replace revocation checks

RQ3: The Real-World Threat Surfaces

❖ **Misleading Certificate Transparency (CT) Monitoring**

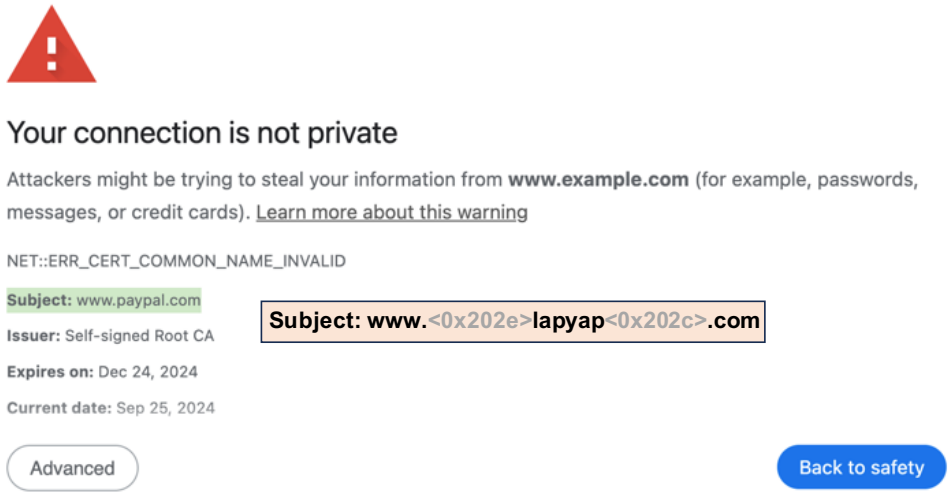
Malicious or compromised CAs can exploit parsing inconsistencies to make CT monitors miss forged certificates, allowing the concealment of deceptive identities.

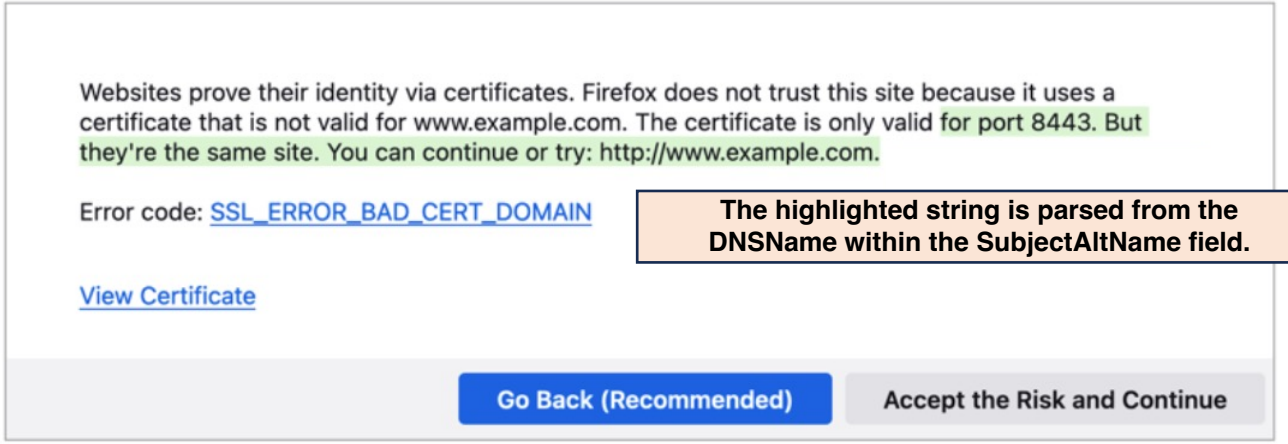


RQ3: The Real-World Threat Surfaces

❖ User Spoofing

Specially crafted certificate fields can manipulate how browser warning pages are displayed, potentially tricking users into trusting unverified sites, though practical exploitation requires stringent conditions and would have limited impact.





Chrome case: the warning page renders bidirectional characters in the Subject CommonName field.

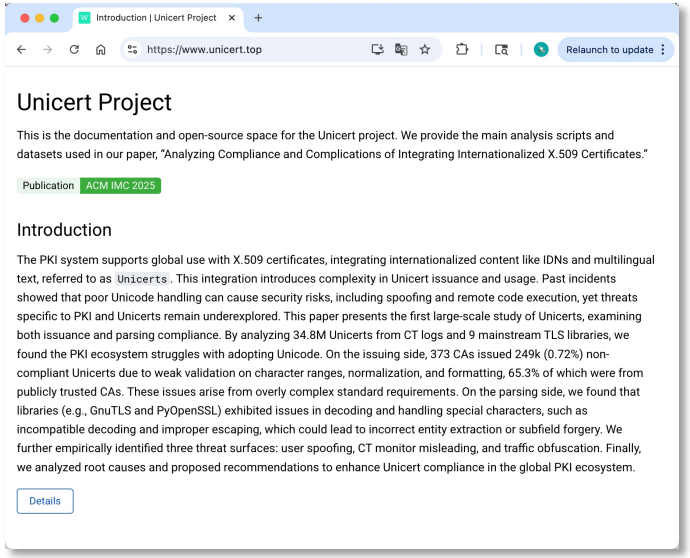
Firefox case: the page could show misleading information derived from a malformed SubjectAltName (SAN) field.

Conclusion, Mitigation, and Contributions

- **Summary**
  - Achieving a truly internationalized PKI is currently challenging due to systemic issuance noncompliance and universal parsing flaws.
- **Key takeaways and current consensus**
  1. Building an internationalized PKI requires collaboration among CAs, developers, and standards bodies.
  2. Unicode integration spans evolving standards, while current CA/B BRs lag behind updates (e.g., IDNA 2008).
  3. Correct ASN.1 parsing in Unicerts is as essential as compliant issuance.

Conclusion, Mitigation, and Contributions

- **Our Contributions:**
  - First systematic study on Unicert issuance and parsing compliance.
  - Identification and large-scale prevalence measurement of 3 types of issuance noncompliance.
  - Empirical discovery of decoding and character handling anomalies in all 9 tested TLS libraries.
- **Mitigation**
  - We responsibly disclosed the issues and provided checking rules, tools, and recommendations to affected CAs and TLS library teams; coordinated on fixes.
  - We released our testing tools for future work and more discussions with the community



Project website: https://www.unicert.top

Thank you!

Q&A

Presenter: **Mingming Zhang**

Contacts:
zhangmm717@gmail.com

Paper available at:
https://www.unicert.top
https://doi.org/10.1145/3730567.3764483