



DA^名LI NAME

“Identity”-Based Cryptography

What it is NOT

+

Where is Money?

Lessons learned from near
half century for
Public Key Cryptography

DaoliName
January 2019



Executive Summary

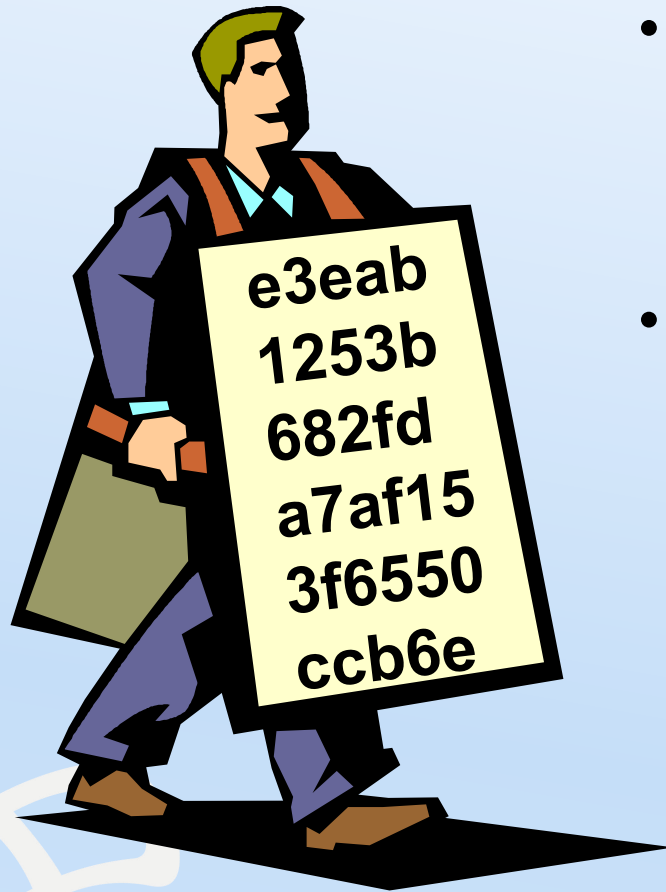
- When public key crypto was invented, offline authentication was a magic; now everyone hates offline
- When ID-based crypto was invented, non-interaction was a magic, even though the cost for communications was already much lowered
- In an era everybody is online, public key should feature online too, and even ID based
- Domain Name Systems (DNS) as an extremely efficient and successful online servicing hierarchy, can be streamline secured if domain names are public keys
- PKC2.0: Registered bitstrings as authenticated public keys to enable crypto capabilities for smart dusts

Content

- Not very interesting technicalities on Identity + “Identity” Based Cryptography (IBC + “IBC”)
- Some slides, in particular those on IBCs, were pasted from SlideShare; we did so in order to walk ourselves away from pitfalls, historical ones even backdate to PKI
- Bla, bla, bla, ..., however none is PPT technology
- Jump to Slide entitled: “Fortunately Internet was NOT Invented by Cryptographers”, that should be a good way to review this slide set
- Money is never an issue, so technical reviewers do feel free to skip anywhere money is mentioned :-)

Footnote: “Identity”, “ID”, “IBC”, “IBE”, “IBS”: Quotes to eye-catch a key and unique difference from all known Identity Based Crypto: We feature *online & interaction*

Problems with PKI



- Sender (or verifier) must have recipient's (or signer's) certificate
- Complexity of certificate mgnt; A real world hard truth: Daily-in-use public key certificates bind “Crypt32.dll” as “chained-melody” to the Chinese Patent Office website, forcing 2019 users to re-install WindowsXP clients in order to run 32-bit MS iExplorer (to sign patent filings)

Identity-Based Cryptography (IBC)



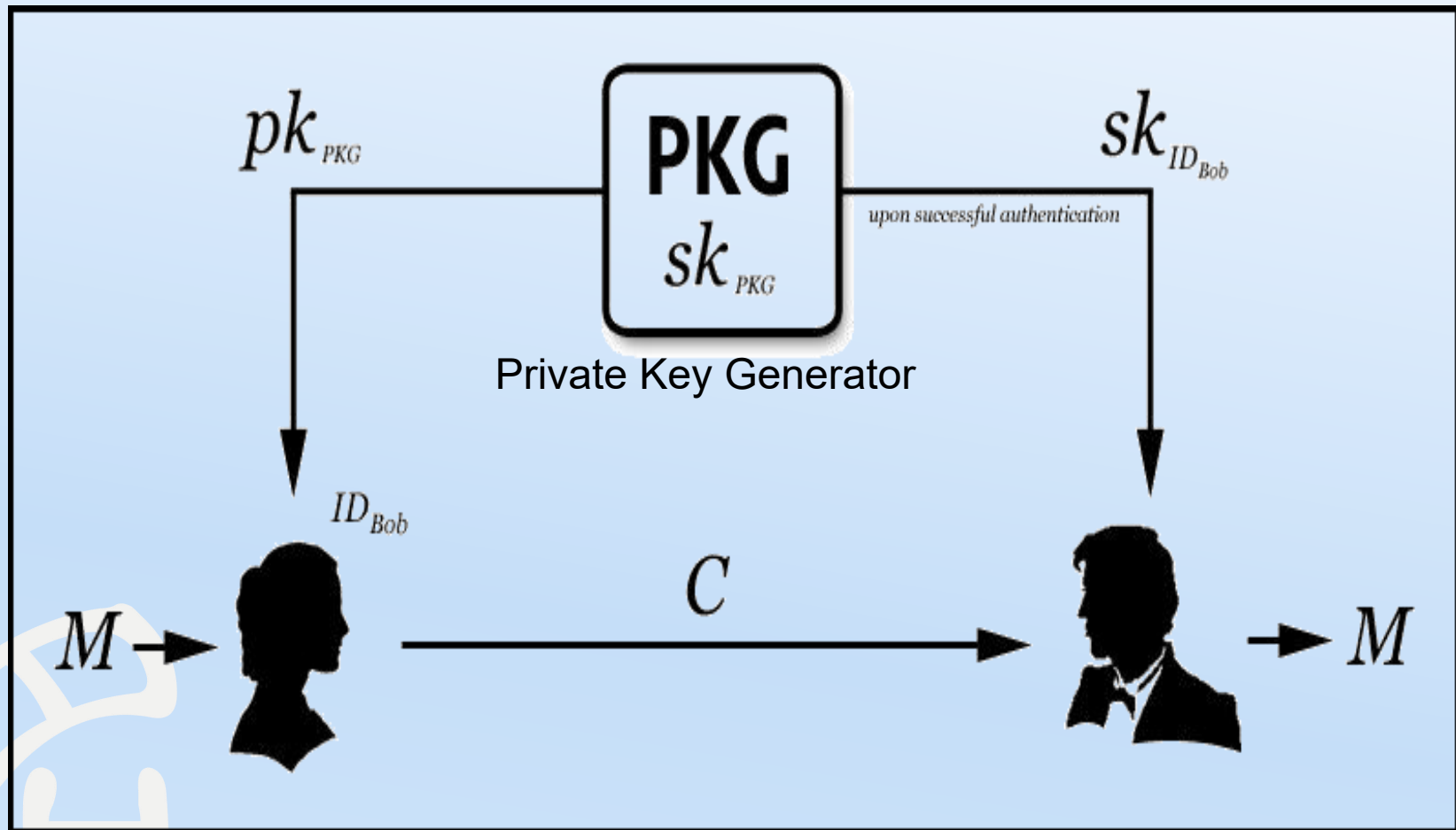
- *Non-Interactive:*
Cryptography for unprepared users (“Unprepared” is not very meaningful. No surprise please.)
- Public key is the user's identity, e.g., email address, phone number, ... (We like domain names, IP addresses, phone numbers, ... to be public keys.)
- Sender only needs to know recipient's identity to send an encrypted message (How?)

History of ID Based Crypto

- Seminally proposed by Shamir in 1984
Shamir came up with a working system for identity-based signature (IBS), but no system for identity-based encryption (IBE)
- First IBE system discovered in 2001 by Boneh and Franklin, using Weil pairing
- Many many schemes followed, but few application, unfortunately



Identity-Based Crypto (IBC)



“Fairy Tales” About IBC

When IBC was an ugly duckling,
how could it avoid merciless taunts?

1. Inherent key escrow
2. No key revocation
3. PKG requires extremely strong security assurance, since it holds all private keys and must remain online

Bla bla bla ...

None of these accusations make a good sense

We shall refute all and each one by one

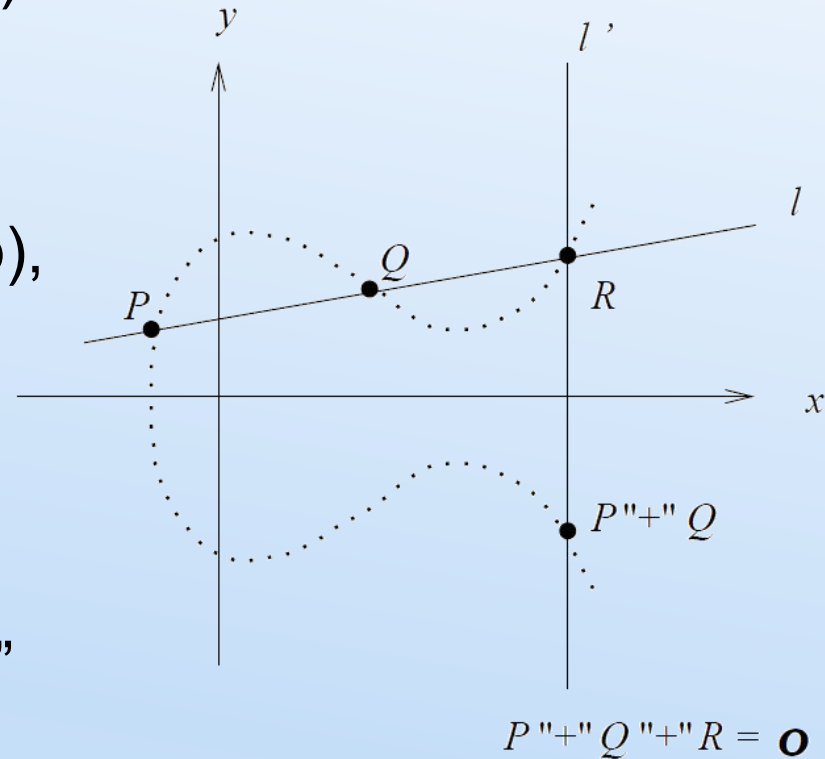
Elliptic Curve Additive Group

$$y^2 = x^3 + ax + b \pmod{p}$$

- x, y, a, b in $GF(p)$

If $4a^3 + 27b^2 \neq 0 \pmod{p}$,
then a group can be
formed:

- Points on the curve
plus the infinity point
- Additive operation “+”
being geometrically
defined



Bilinear Pairing: 1st Crypto Use

Burton Kaliski Jr., 1988

G, G_1 : (sub)groups of prime order $q \mid (p-1)$

Bilinear pairing map \hat{e} : let $P = (x, y) \in G^2$

- Bilinear: $\hat{e}([a]P, [b]P) = \hat{e}(P, P)^{ab}$
- Non-degenerate:
 P generates $G \Rightarrow \hat{e}(P, P)$ generates G_1
- Efficiently computable



Kaliski's Recounting the History: Victor Miller visited Ron Rivest when I was a graduate student, and he met with me about my research. If I recall correctly, I asked him if he knew a way to determine whether an elliptic curve group was cyclic, and he suggested the Weil pairing. He also gave me a copy of his algorithm for computing the Weil pairing, and agreed that I could implement it for my thesis.

Diffie-Hellman Problem

Computation: Hard; Decision: Easy



Given $\langle P, [a]P, [b]P, [c]P \rangle$ for integers a, b, c

Deciding $c \stackrel{?}{=} ab \pmod{q}$ use bilinear pairing:

$$\hat{e}(P, [c]P) \stackrel{?}{=} \hat{e}([a]P, [b]P)$$

Since this is equivalent to

$$\hat{e}(P, P)^c \stackrel{?}{=} \hat{e}(P, P)^{ab}$$

Evaluation: No need of knowing integers a, b, c

This is extremely important and useful for IBC

System Setup

For primes p, q and groups defined in slides 9, 10, let a Private Key Generator, PKG, set an arbitrary curve point in the order q subgroup:

$$P = (x, y) \bmod p$$

PKG picks a random integer $s < q$, and sets $P_{\text{PKG}} = [s]P$

PKG chooses two hash functions:

$G: \{0,1\}^* \rightarrow (x, y)$: hash arbitrary bitstring to the curve

$H: \hat{e}(P, Q) \rightarrow \{0,1\}^{\log_2 q}$: hash $GF(p)$ to $\log_2 q$ -bit strings

PKG publicizes system public parameters:

$$\langle p, q, P, P_{\text{PKG}}, G, H \rangle$$

PKG securely keeps secret “master_secret_key”: s

User Key Setting

Alice's key setting:

Her public key: $R_A = G(\text{Alice})$

Her private key is issued by PKG as: $d_A = [s]R_A$

Note: $R_A = (x, y) \bmod p$, is a point on the curve

N.B. Hashing to elliptic curve is very very important!
Some authors, e.g., those who have proposed SM9,
choose to hash to discrete logarithm integers

We shall comment undesirable consequences in so
doing in a moment

ElGamal Encryption

Encryption: To send encrypted M to Alice,
Bob picks a random integer $r < q$, and computes

$$C = \langle [r]P, M \oplus H(g_A^r) \rangle$$

where $g_A = \hat{e}(P_{PKG}, R_A)$

Decryption: (for $C = \langle U, V \rangle$)

$$V \oplus H(\hat{e}(U, [s]R_A)) = M$$

Decryption correctness:

$$\hat{e}(U, d_A) = \hat{e}([r]P, [s]R_A) =$$

$$\hat{e}([s]P, [r]R_A) = \hat{e}(P_{PKG}, R_A)^r = g_A^r$$

That is, Alice can use her private key to reproduce g_A^r ,
equal to Bob computed using Alice's IBC public key

Fairy Tale Refutation: “Inherent Key Escrow”

Let PKG2 publicize point $Q_{PKG2} = [t]P$, for a random integer $t < q$ as its “master” secret key

For $R_A = G(\text{Alice})$, her private key $d_A = [s]R_A + [t]R_A$

Then, to non-escrow ElGamal encrypt message M :

Bob picks a random integer $r < q$, and computes:

$C = \langle [r]P, M \oplus H(g_A^r) \rangle$, where

$g_A^r = \hat{e}(P_{PKG}, R_A)^r * \hat{e}(Q_{PKG2}, R_A)^r$ Bob computes

$= \hat{e}([r]P, d_A)$ Alice computes in decryption

A Stronger Refutation

Let Alice play the role of “PKG2”: $Q_A = [a]P$,
for $a < q$ being her “private_key_self”, *kept secret from*
PKG, Alice's “private_key_whole” is: $d_A = [s]R_A + [a]R_A$

Then, to no-key-escrow ElGamal encrypt message M ,
Bob picks a random integer $r < q$, and computes:

$$C = \langle [r]P, M \oplus H(g_A^r) \rangle, \text{ where}$$

$$g_A^r = \hat{e}(P_{PKG}, R_A)^r * \hat{e}(Q_A, R_A)^r \quad \text{Bob computes}$$

$$= \hat{e}([r]P, d_A) \quad \text{Alice computes in decryption}$$

This is an “IBE”, quoted as Bob needs to first get Alice's
non-ID based, *yet authenticated*, public key Q_A , we will
discuss absurdity for PKG to forge Q_A in a backup slide

Fairy Tale Refutation: “No Key Revocation”

Lift version first:

To interact or not to
interact?

That is THE question!

Interaction can let Bob get
non-ID based, *yet well
authenticated*, public key
 Q_A of Alice, and provide a
new usefulness to “IBC”

Rationale unfolding ...



History of Digital Signature

In 1976, Diffie-Hellman Open Question:
How to digitally sign a document, use or not use
(g , g^x) form of public key

In 1978, RSA answered so beautifully:
 $M^d \bmod N$

In 1984, ElGamal answered for public key being
(g , g^x): use a random r to cover the private key a ,
the covered discrete log can be computed publicly

An “ID” Based Signature

Let Alice’s “private_key_PKG” be:

$$d_A = [s]R_A = [s]G(\text{Alice})$$

Let Alice also have “private_key_self”: $a < q$, kept secret from PKG as in no-key-escrow encryption

In registration, Alice signs the system public point P :

$$Q_A = [a]P$$

Alice's public key parameter:

$$\langle R_A, Q_A \rangle$$

“ID” quoted as the public key has a *non-ID feature* Q_A

An “ID” Based Signature

Alice to sign M :

$$\text{Sig}_A(M) = d_A + [a]G(M)$$

Public Verification:

$$\hat{e}(P, \text{Sig}_A(M)) \stackrel{?}{=} \hat{e}(P_{\text{PKG}}, R_A) * \hat{e}(Q_A, G(M))$$

Staring at Alice's “private_key_self” longer:

- Integer a can be one-time and random, and thus, $Q_A = [a]P$ becomes part of randomized signature
- After signing with a , Alice can further sign a next signature using $a + b \pmod{q}$, with randomized signature part being $Q'_A = [a + b]P$, and so on ...

Tale Refutation for Signature: “Inherent Key Escrow”



- Alice can publicize first $Q_A = [a]P$, where the first “private_key_self” a is kept secret from any PKG; she can use $a + b$ as new “private_key_self”, for $Q'_A = [b]P$, let verifier use $Q_A + Q'_A$ in place of Q_A
- Alice's “private_key_PKG” d_A can also easily be issued by a distributed multiple of PKGs, exactly the same as refutation to “inherent key escrow” tale for ElGamal encryption; our usecase does feature so
- A verifier can also prevent PKGs from forgery by challenging a multiplier to “private_key_self” of Alice (detail provided in a backup slide)

Refutation to Fairy Tale 3: Two Approaches to IBC

1. Hash to elliptic curve

$G(\text{Alice})$, $G(M)$, are curve points (x, y) to make use of beautiful ease of decision DH

2. Hash to discrete logarithm (e.g., SM9)

Signature follows ElGamal's old game of using a random number to protect private key

Our “IBC” schemes all enjoy approach 1: PKGs can be distributed, even Alice be one such, to refute Fairy Tale 3: “Highly strong security requirement for PKG”

IBCs taking approach 2 seem to suffer all tale scorns

It's Open System, Stupid

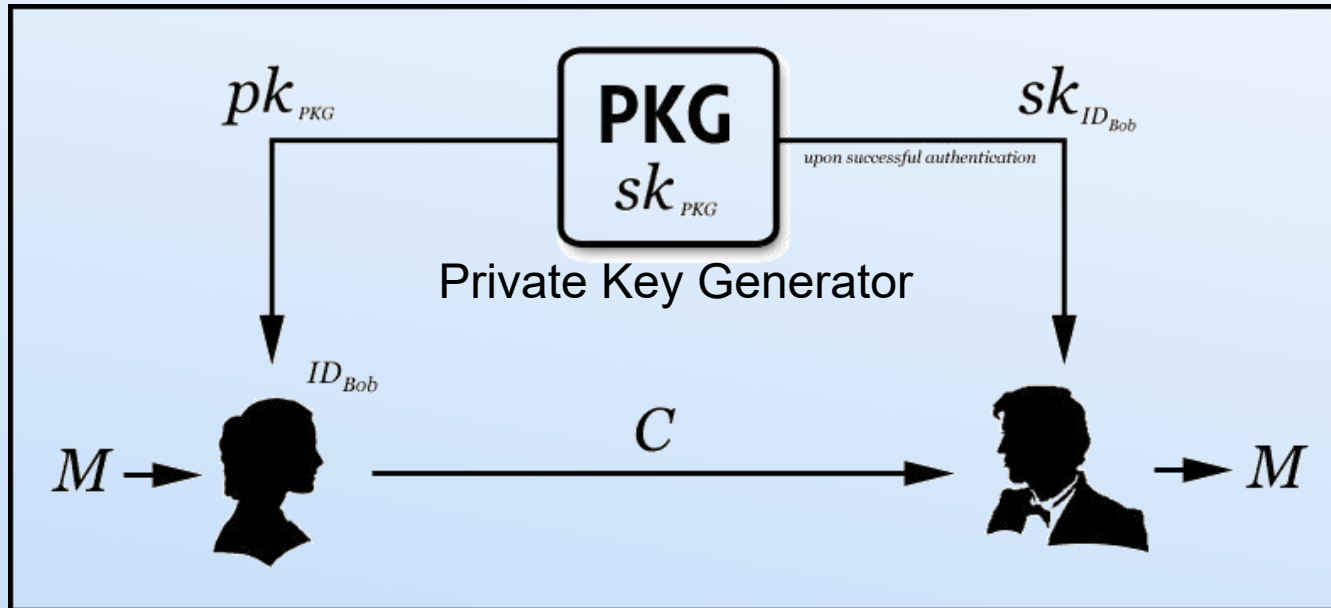
Voltage Security Inc. (Boneh's company)
to sell ID-Based Crypto Email

Many (more likely, cryptographers) think Email is
the killer-app of ID-Based Crypto

Unfortunately, this killer-app is itself now almost
killed, e.g., colleagues & friends of DaoliName
have not been using it for ages

At any rate, even Email must also be an open
system, mustn't it?

Cryptographers' System View



Viral growth of a product/service/business by opening system up shouldn't be cryptographers' job optimum

A crypto product should always be “hush-hush”
Make it open? What are you talking about?!

Fortunately Internet was NOT Invented by Cryptographers


Internet has never been architect-ed to be a product, or for an app, not even for an industry, Internet is an **Open System!**

Crypto if vended as product (box) can hardly be an open system

What is easily missing? Key management

Imagine Alice having bought an “IBC Email” box:
“Has Bob also bought this four-double-letter fragment named gadget from Dan?” or “Is Bob's ID based key still OK?”

PKI Certificate as a Product

Given key revocation can hardly be a sold box, why to vend PKI certificates in the first place? Our WindowsXP, 32-bit iExplorer using Chinese patent filers are only 1 drop in an ocean of govt services users who may be experiencing mental, if not physical, fragmentation 

Making PKI an Open System shouldn't be very difficult; it's the **thought** got stuck offline that is!

An offline box is “buy once use forever” (BOUF). To vend smart BOUF, money is not enough to hire smart makers. Now nearly century/2 elapsed: Who pays for key revocation to have user experience cared for?

Has Public Key Ever Well Done Key Management?



1976: Key agreement problem was suddenly solved,
key channel setup no longer need Kerberos-like online

1978: Offline CA PKI, then you go!

2000: Even OFF offline was discovered: ID is public key,
not even communication needed!

1997 (RFC 2065) to date: DNSSEC, Extremely efficient
online DNS hierarchy is bundled to same structured but
offline PKI hierarchy. Can, on Gods' green Earth, pull-
the-plug take so long no thanking to PKI burning energy?

2019: Everybody ***hates*** offline!

Domain Name System DNS: An Awesome Online Service



Internet trade-mark registration of names +
online query for (registered name, IP addresses)
binding

Real-time query, if (DN, IP, TTL) with TTL being valid,
use the binding, otherwise, as if binding revoked

Both being pyramid hierarchies, online DNS is far far
more successful than offline CA PKI

It's Open, cross app, and cross industry!

DNS will be with us forever: Programmers can't hard-
wire code IPs

Bitstring Registration for Streamline Key Mgmt Services



Global authenticated bitstring registration: “Is this phone no., (IP addr, ...) registered as a public key?”

Use the existing DNS “trade-mark” reg hierarchy

Answer is small (see a backup slide) to be wrapped in a 512-byte UDP packet, not only to penetrate firewalls, but more importantly to provide **value** to client users: streamline key mgmt service for dusts

User: “Is the private key behind this bitstring (DN, IP addr, phone no) crypto-worthy (e.g., not revoked)?”

DaoliName: “Check TTL yourself! It's 'IBC' signed!”

Where IS Money?

PKI has never answered this question well,
in near half century proud of offline “magic” authentication

Worse for IBC: PKI can sell some certificates, i.e., service
We've never heard IBC made any servicing money

Open up online, like the Internet, is the key to success

From data plane: Billions of smart dusts shall appreciate
value of their “IBC” authenticated bitstring IDs to enjoy
crypto, e.g., an IP addr can run IPSec (reverse DN query)

From control plane: A new TLD “.mi” (sounding crypto in
Chinese) to sell *secure* name registration, as “IBC”
signature secured DNS (volume of DN squatting probable)

What for Sale

The following two well established services have been very lucrative:

1. Trade-mark registration is a brand name (maybe value add on, + logo binding) service, to sell, physically, to a nation's businesses
2. DNS registration = Service 1 + IP address binding service, to sell, *digitally*, to the globe's businesses

The new “IBC” enabled bitstring registration = Service 2 + authenticated public key capability service, to sell, digitally and *securely*, to the globe's businesses + individual users + client dusts. Securely means authenticated bitstring as public key bound to, e.g., IP addresses

Summary

Historical rethinking of (not quoted, aka non-interact or offline) ID Based Crypto, and PKI

Refutation to all nonsensical tales about IBC:

If you can't fix it, feature it!

Crypto, in fact, key mgnt, vended as boxes, being too smart, too conservative, too complex, and too fragment, to be early to market, vending crypto boxes is hard, no money, no progress, then chunks of century go by

Online is THE feature, never a problem

“IBC” is not non-interactive, not offline, not a product, not for one industry. It is services for **PKC 2.0**

Backup Slides: “IBC” Signature Authenticated Bitstrings



A registered bitstring suffices to provide public keys to verify its bona-fide-ness as being signed authoritative answer

Upper level hierarchical nodes: delegated & distributed PKGs,
Commercial lower level nodes: authoritative servers

Upper level nodes colluding forging a non-ID point Q_A of a low level node is as absurd as PKI CAs faking user certificates

“IBC” signatures of upper level nodes form “Compressed Fingerprints” (CF) of theirs down the DNS hierarchy, and these CFs are curve “+”ed to a single curve point

Final CF small enough (Barreto-Naehrig curves) for 512-byte UDP packet to penetrate firewalls to reach clients & dusts

“IBC” CFs Secured DNSSEC Feature Online and Interaction



A DNS Asker, e.g., client or ISP on behalf of client, as CF verifier, queries DN to the DNS hierarchy

Each of the answering nodes upper in the DNS hierarchy above a leaf node Auth (who has “authoritative record binding”), computes a CF using the “IBC” signature formula to answer Asker

Asker challenges Auth random v , requiring Auth to use va as “private_key_self” to randomize CF_{Auth} output

Asker verifies: $\hat{e}(P, CF_{Auth}(M)) = \dots * \hat{e}(Q_{Auth}, G(M))^v$

Biz Plan 1: Cold Start for DNSSEC

Sales to DNS Servers

- “IBC” secured DNSSEC registration, to banks, govt services and organizations; these entities have force to impose clients to use “IBC” signed DNSSEC, to grow to commercial organizations
- Hardware Signature Modules (HSMs) running PCT patented “IBC” Compressed Fingerprint protocols, to DNS servers. This sales will be virally growing to the global wide millions of authoritative DNS commercial servers (lower level nodes)

Not to clients or IoT dusts (too fragmented market):

- Open source reference code plugins for mobile apps, then for web browsers, OS drivers ... to virally spread

Biz Plan 1: Go to Market Strategy

Top level non-commercial nodes, i.e., root “”, g-TLDs, cc-TLDs (“.com”, “.net”, “.org”, “.edu”, ..., “.uk”, “.cn”, ...), are presumably hard to join “Cold Start”. Thus:

- Responsibly build *their* CFs for them. Responsibility means to very carefully secure *their* “private_key_self”s, and respectively *their* CFs, securely kept in HSMs on *their* behalf
- Sell DNSSEC registration to commercial levels of nodes as descendants of the *unsold* top level non-registered names
- IETF RFCs, open source recursive resolvers, to make viral spreading clients. If CF verification returns NO, re-query, e.g., DaoliName servers, for CF signed authoritative bindings
- Someday, top level nodes would appreciate value of “IBC”. Then *Their* “private_key_self”s and CFs could be *returned* to them as *their* due properties

Biz Plan 2: Dream of A Cryptographer Turned Engineer

All smart phones, IP addresses, smart dusts, in particular, those appearing 5G ones, and IPv6 route-able devices, may have cross app, cross industry, cross social media, ..., **PKC 2.0**

Alice's query: Is this unique phone number (or IP address) of Bob (of course should be anonymous) registered with authentication to have “ID” Based Crypto capability?

“IBC” signed answer can just use the existing 512-byte UDP packet to transmit! No reinvention of EDNS0, or TCP DNS, or any such new wheels!

Open System Open Standard Open Source Open IETF RFCs

One World, One Dream

Only the World First to become better and better,
can China be also good as part of :-)

We want to work with the World to secure DNS, and
to pervade **PKC 2.0**

Money is never an issue, it can only be great return!



DA  LI NAME