Linkable Message Tagging

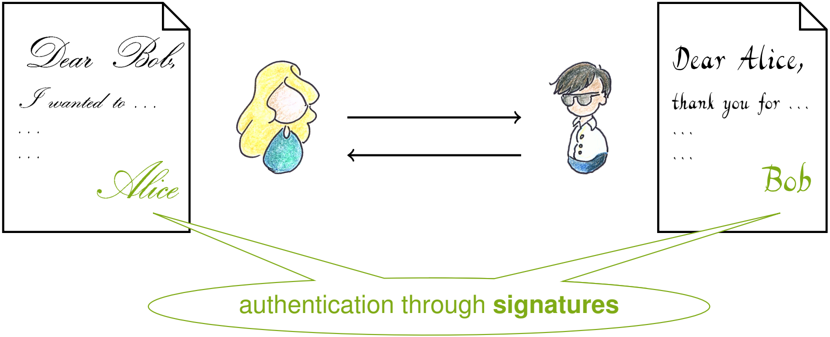
Solving the Key Distribution Problem of Signature Schemes

Felix Günther
Technische Universität Darmstadt
Germany

Bertram Poettering
Ruhr-Universität Bochum
Germany

Sending Letters

back in ancient times...



drawings by *Giorgia Azzurra Marson*

Sending Letters

today...

Dear Bob,

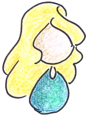
I wanted to ...

...

...

Alice

sk_A



sk_B

Dear Alice,

thank you for ...

...

...

Bob

$\sigma_A \leftarrow \text{Sign}(sk_A, \dots)$ $\sigma_B \leftarrow \text{Sign}(sk_B, \dots)$

authentication through **digital signatures**

Sending Letters

today…

TECHNISCHE UNIVERSITÄT DARMSTADT

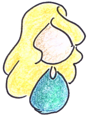
```
Dear Bob,
I wanted to ...
...
...
           Alice
```

But: Alice and Bob might not even know each other prior to communicating

sk_A, vk_B sk_B, vk_A

```
Dear Alice,
thank you for ...
...
...
             Bob
```

sk_A, vk_B



sk_B, vk_A

$\sigma_A \leftarrow \mathrm{Sign}(sk_A, \ldots)$

$\mathrm{Verify}(vk_B, \ldots, \sigma_B)$

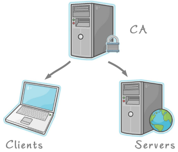
$\sigma_B \leftarrow \mathrm{Sign}(sk_B, \ldots)$

$\mathrm{Verify}(vk_A, \ldots, \sigma_A)$

How to authentically distribute keys?

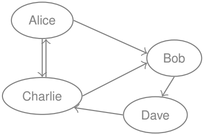
Approaches So Far
(Selection)

(Hierarchical) PKIs



- (X.509) certificates issued by CAs bind keys to identities
- HTTPS-secured web, S/MIME email encryption/signing
- large number of (trusted) root and intermediate CAs
- unclear trust relations / CA compromises (DigiNotar, TURKTRUST, . . .)
- revocation seems difficult

(Social) PKIs



- web of trust, personally signing keys
- OpenPGP
- scalability, time-consuming/error-prone authentication ('key signing parties')
- privacy issues (reveals social relationships)

Approaches So Far
(More Academic Selection)

TECHNISCHE UNIVERSITÄT DARMSTADT

Identity-Based Signatures (Shamir 1984)

- public key = identitiy of a user (e.g., vk_A = `"Alice"`)
- inherent key escrow problem (master key which can decrypt everything)

Certificateless Signatures (Al-Riyami, Paterson 2003)

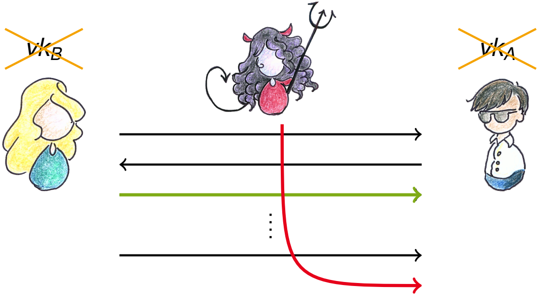
- hybrid between PKI and identity-based
- user obtains *partial* private key to complete on her own
- still requires some trust in (and existence of) central party

Message Recognition (Weimerskirch, Westhoff 2003)

- method to recognize each others' messages as authentic
- requires prior exchange of small amount of authentic data

A Novel Approach:
History-Based Message Authentication

TECHNISCHE UNIVERSITÄT DARMSTADT



Goals

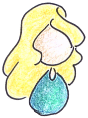
- detect forged messages
- given a single authentically delivered message (unknown which one it is)
- without explicit exchange of verification keys

New tool: Linkable Message Tagging

Linkable Message Tagging

Syntax

tk_A



$\tau_1 \leftarrow \text{Tag}(tk_A, m_1)$ —— m_1, τ_1 ——→

$\tau_2 \leftarrow \text{Tag}(tk_A, m_2)$ —— m_2, τ_2 ——→

⋮

$\tau_n \leftarrow \text{Tag}(tk_A, m_n)$ —— m_n, τ_n ——→ $\text{Link}(m_1, \tau_1, m_n, \tau_n) = 1$

LMT Scheme

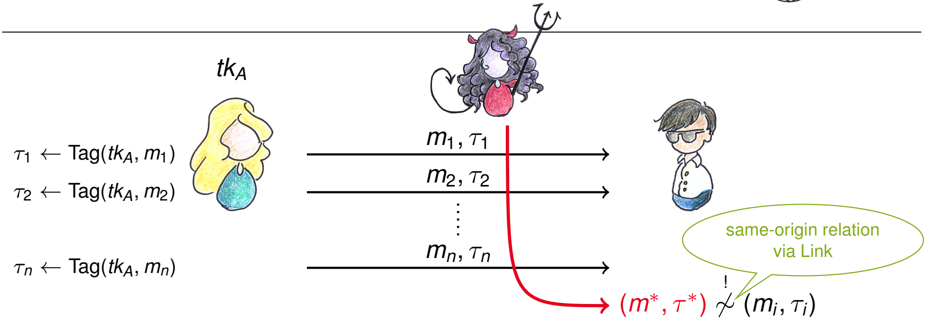
- $\text{KGen}(1^\lambda)$: Generate a tagging key tk.
- $\text{Tag}(tk, m)$: Output a tag τ for a message m.
- $\text{Link}(m_1, \tau_1, m_2, \tau_2)$: Output 0 or 1. (short for 1: $(m_1, \tau_1) \sim (m_2, \tau_2)$)

  required to be an equivalence relation

Intuition: 1 iff (m, τ)-pairs created with same key.

Linkable Message Tagging

Security



(Existential) Unforgeability

- Adversary seeing tags τ_i for messages m_i of its choice
- is not able to forge a new tag τ^* for an unseen message m^*
- such that $(m^*, \tau^*) \sim (m_i, \tau_i)$ for any (m_i, τ_i).

- strong unforgeability: τ^* can be for a previously seen message m_i

Linkable Message Tagging
Envisioned Application

Envisioned application: automated email authentication

- easy-to-use and fully-automated cryptographic authentication of email
- automatically set up tagging keys (on first use)
- automatically tag all outgoing emails
- automatically visually group incoming emails (according to relation $\sim$)
- advantages:
  - everything fully automatic (no user interaction required)
  - no exchange of verification keys needed
- unforgeability guarantees: adversarial emails are grouped separately

TECHNISCHE UNIVERSITÄT DARMSTADT

BLS-LMT scheme

based on BLS signatures (Boneh, Lynn, Shacham 2001)

- Ingredients:
  - (symmetric) bilinear group $\mathbb{G} = \langle g \rangle$ (prime order q) with map $e \colon \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$
  - hash function $H \colon \{0,1\}^* \to \mathbb{G} \setminus \{1\}$
- KGen(1^λ): $x \xleftarrow{\$} \mathbb{Z}_q$, output $tk = x$.
- Tag(tk, m): Output a $\tau = H(m)^{tk} = H(m)^x$.
- Link(m_1, τ_1, m_2, τ_2): Output 1 if $e(H(m_1), \tau_2) = e(H(m_2), \tau_1)$.
- Correctness: (in particular Link establishes equivalence relation)

$$(m_1, \tau_1) \sim (m_2, \tau_2) \iff e(H(m_1), H(m_2))^{tk_2} = e(H(m_2), H(m_1))^{tk_1} \iff tk_1 = tk_2$$

- Security: BLS-LMT is strongly unforgeable if CDH is hard in $\mathbb{G}$, in the ROM

(proof via strong unforgeability of BLS signatures)

Linkable Message Tagging

Generic Relation with Signatures

TECHNISCHE UNIVERSITÄT DARMSTADT

- recall: LMT is not a public key primitive!
- natural and efficient transformations between LMT and signature schemes
  - + perhaps surprising, interesting theoretical relation
  - − little hope for practical construction from symmetric primitives only

Signature $\longrightarrow$ LMT

- basic idea: use signing key as *tk* and include verification key in tag: $\tau = (\sigma, vk)$
- several design choices for admissible equivalence relations defined by Link
- inherits signature scheme's (existential/strong) unforgeability

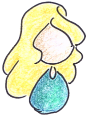
LMT $\longrightarrow$ Signature

- basic idea: use *tk* as *sk* and distinct tag as verification key: $vk = \text{Tag}(tk, \texttt{"0"})$
- signature verification through Link-ing with verification key
- again preserves (existential/strong) unforgeability

Automated Email Authentication Revisited

TECHNISCHE UNIVERSITÄT DARMSTADT

tk_A



$\tau_1 \leftarrow \mathrm{Tag}(tk_A, m_1)$ $\xrightarrow{m_1, \tau_1}$

$\tau_2 \leftarrow \mathrm{Tag}(tk_A, m_2)$ $\xrightarrow{m_2, \tau_2}$

$\vdots$

$\tau_n \leftarrow \mathrm{Tag}(tk_A, m_n)$ $\xrightarrow{m_n, \tau_n}$ $\mathrm{Link}(m_1, \tau_1, m_n, \tau_n) = 1$

in envisioned automated email authentication:
Link must be checked with (m, τ) from each origin/$\sim$-group

—

would be nice if origin was efficiently identifiable

Classifiable Message Tagging

Syntax

tk_A

$\tau_1 \leftarrow \mathsf{Tag}(tk_A, m_1)$ —— m_1, τ_1 ——→

$\tau_2 \leftarrow \mathsf{Tag}(tk_A, m_2)$ —— m_2, τ_2 ——→

⋮

$\tau_n \leftarrow \mathsf{Tag}(tk_A, m_n)$ —— m_n, τ_n ——→ $\mathsf{Classify}(m_n, \tau_n) = cid$

CMT Scheme

- $\mathsf{KGen}(1^\lambda)$: Generate a tagging key tk.
- $\mathsf{Tag}(tk, m)$: Output a tag τ for a message m.
- $\mathsf{Classify}(m, \tau)$: Output a class identifier cid.

  Intuition: each tk corresponds with one specific cid_{tk}.

- (existential/strong) unforgeability defined as expected

Classifiable Message Tagging

Generic Relations

TECHNISCHE UNIVERSITÄT DARMSTADT

CMT schemes are special LMT schemes

- by defining: $\mathsf{Link}(m_1, \tau_1, m_2, \tau_2) = 1 \iff \mathsf{Classify}(m_1, \tau_1) = \mathsf{Classify}(m_2, \tau_2)$
- but not all LMT schemes have CMT analogues
- e.g., BLS-LMT: cid_{tk} could be tk or g^{tk}, contradicting DLP/CDH

Signature $\longrightarrow$ CMT

- again: use signing key as tk and include verification key in tag: $\tau = (\sigma, vk)$
- use class identifier $cid = vk$

CMT $\longrightarrow$ Signature

- use class identifier as verification key $vk = cid_{tk}$

Schnorr-CMT scheme

based on Schnorr signatures (Schnorr 1990)

- Key insight: Schnorr *vk* can be reconstructed from any valid signature

- KGen(1^λ): *tk* = Schnorr signing key
- Tag(*tk*, *m*): τ = Schnorr signature
- Classify(*m*, τ): Output *cid* = Schnorr verification key, reconstructed from τ

- Security: Schnorr-CMT is strongly unforgeable if DLP is hard, in the ROM

  (proof via strong unforgeability of Schnorr signatures)

- Efficiency: $\approx$ 50,000 classifications/sec on a current high-end CPU
  using elliptic-curve-based `Ed25519` (Bernstein et al. 2011)

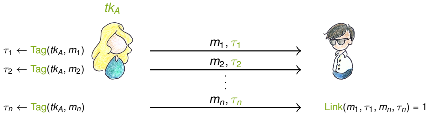
Summary

TECHNISCHE UNIVERSITÄT DARMSTADT

History-based message authentication: side-stepping the key distribution problem.

We

- introduce linkable message tagging, authenticating messages without pre-shared verification keys or PKI



- identify the practical subclass of classifiable message tagging
- explore the generic relation between LMT/CMT and signature schemes
- provide efficient constructions

In the **full version** (ePrint 2014/014)

- CMT scheme without random oracles from Waters signatures
- on DSA- and ECDSA-based CMT schemes
- on CMT schemes from Fiat-Shamir transformed signatures
- do S/MIME and OpenPGP lead to efficient CMT schemes?

Thank You!