Non-browser TLS Woes

Dan Boneh

Joint work with M. Georgiev, S. Iyengar,
S. Jana, R. Anubhai, and V. Shmatikov

Proc. ACM CCS 2012

30 second summary

Lots of non-browser systems using TLS:

- Payment gateway SDK, Mobile ads,
  Web services middleware, cloud client API, …

Developers use TLS through a higher level library:

e.g. HttpClient, cURL, Weberknecht, PHP, Python

- **Problem 1**: complex and confusing interfaces to TLS layer
  - result: **lots** of improper server-side cert validation
    ⇒ man in the middle vulnerabilities

- **Problem 2**: little testing of server-side cert. validation
  Should be part of QA testing …

Case studies: (1) PHP / cURL binding

We found four SDKs using it for SSL

PHP TLS binding:

- **fsockopen**("ssl://…") : no server-side cert. validation
- **<u>cURL binding</u>**: cert. verification controlled by

  boolean CURLOPT_SSL_VERIFYPEER

  int CURLOPT_SSL_VERIFYHOST

  0: no server-side cert validation

  1: check the existence of a common name in server cert.

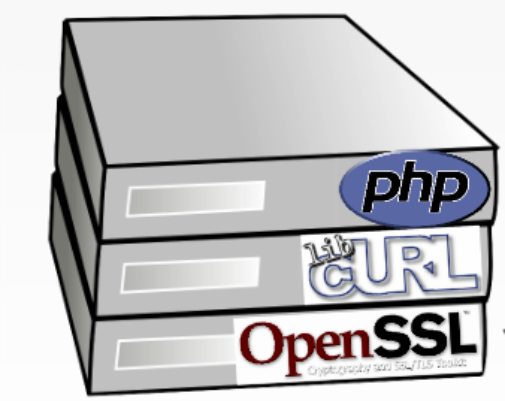
  2: check common name in cert. matches provided hostname

  (default = 2)

Very common mistake:

CURLOPT_SSL_VERIFYPEER = true

CURLOPT_SSL_VERIFYHOST = true



Real World Crypto Workshop, Jan. 2013.

Instead:

Recent Developments in Broadcast Encryption

Dan Boneh

Stanford University

Background: bilinear maps

G , G_2 : finite cyclic groups of prime order q

An admissible bilinear map $e: G \times G \rightarrow G_2$ is:

- Bilinear: $e(g^a, g^b) = e(g,g)^{ab}$ $\forall a,b \in Z$, $g \in G$
- Non-degenerate: g generates G_1 $\Rightarrow$ e(g,g) generates G_2
- Efficiently computable

Several examples where Dlog in G believed to be hard

Many Applications: enc., sigs., NIZK, …

Simplest example: BLS signatures [B-Lynn-Shacham'01]

KeyGen: sk = rand. x in Z_q , pk = $g^x \in G$

Sign(sk, m) $\longrightarrow$ $H(m)^x \in G$ $\qquad e(g, H(m)^x) = e(g^x, H(m))$

verify(pk, m, sig) $\longrightarrow$ accept iff $\qquad e(g, \text{sig}) \stackrel{?}{=} e(pk, H(m))$

Thm: Existentially unforgeable under CDH in the RO model

Beyond bilinear maps: k-linear maps [BS'03]

k-linear map $\quad e: \underbrace{G \times G \times \cdots \times G}_{k} \longrightarrow G_k$ $\quad$ non-degen. & efficient

hard Dlog in G

Even more applications.

Can they be constructed?

k-linear maps: a recent breakthrough

S. Garg, C. Gentry, S. Halevi



<u>Properties</u>: (informal)

- The map $x \longrightarrow g^x$ is randomized
- Representation of $g \in G$ is $O(k)$ bits
- Better than k-linear map: **gradation**

$e_1: G \times G \longrightarrow G_2$

$e_2: G \times G_2 \longrightarrow G_3$

$\vdots$

$e_k: G \times G_k \longrightarrow G_{k+1}$

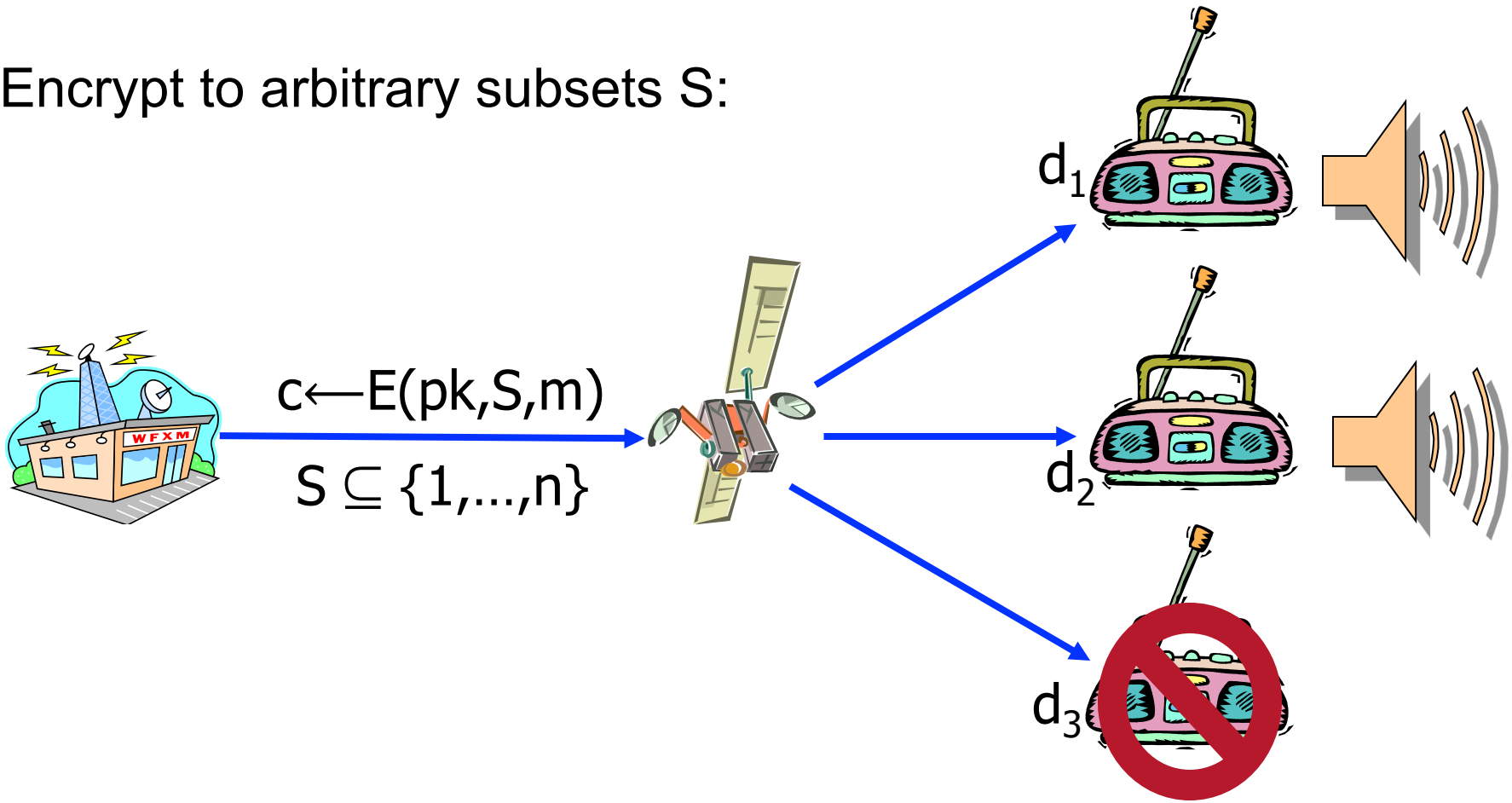
For our purposes:

$e_k: G \times \cdots \times G \longrightarrow G_k$

$e: \; G_k \times G_k \longrightarrow G_{2k}$

Broadcast Encryption [Fiat-Naor 1993]

Encrypt to arbitrary subsets S:



Security goal (informal):

Full collusion resistance: secure even if **all** users in S^c collude

Broadcast Encryption

Public-key BE system:

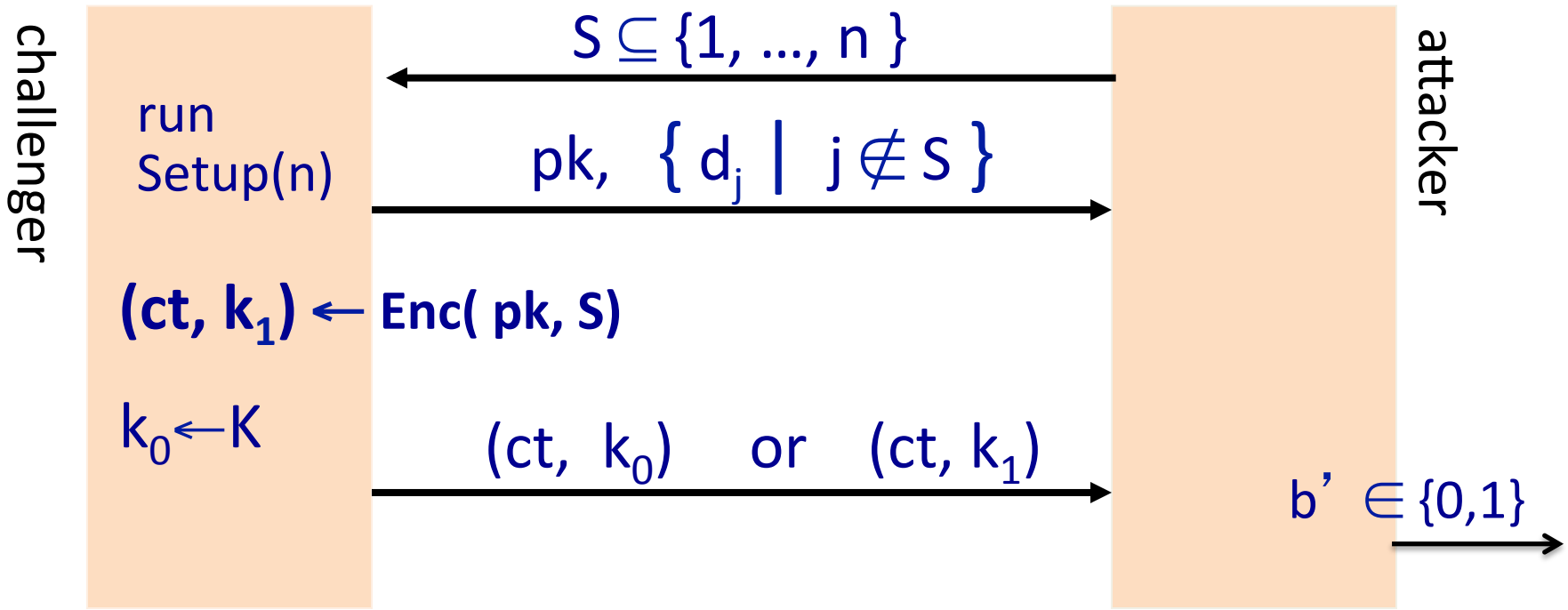
- **Setup**(n) $\longrightarrow$ pub. key **pk,** master sec. key **msk**
- **KeyGen(msk, j)** $\longrightarrow$ d_j (private key for user j)
- **Enc**(pk, S) $\longrightarrow$ **ct** , **k**

  k used to encrypt msg for users $S \subseteq \{1, \ldots, n\}$
- **Dec**(pk, d_j, S, ct): If $j \in S$, output **k**

Broadcast contains ([S], ct , E_{SYM}(k, msg))

Broadcast Encryption: Static Security

Semantic security when <u>users collude</u> (static adversary)

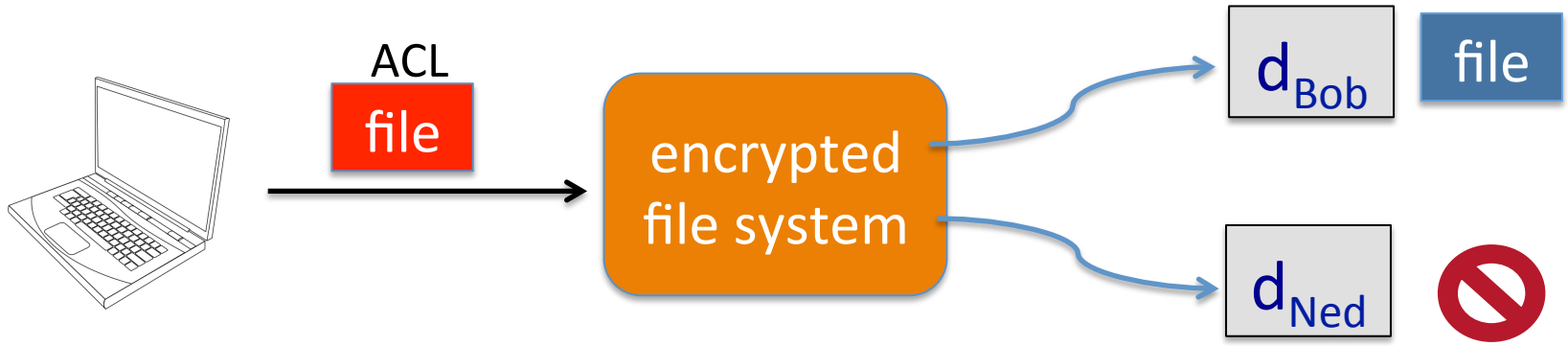


<u>Def</u>: Adv[A] = | Pr[b' is correct] - ½ |

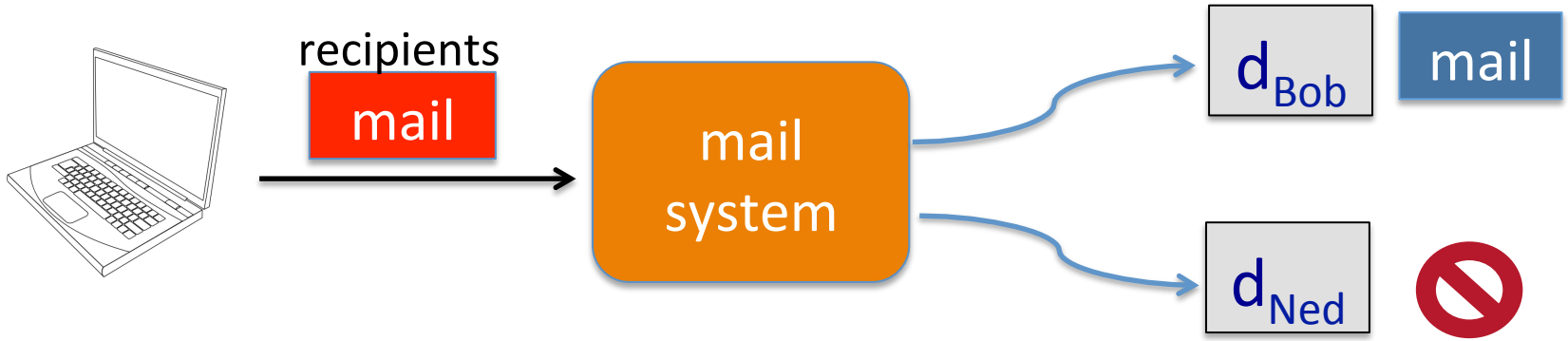
Security: ∀ poly-time A: Adv[A] is negligible

Broadcast systems are everywhere

File sharing in **encrypted file systems** (e.g. EFS):

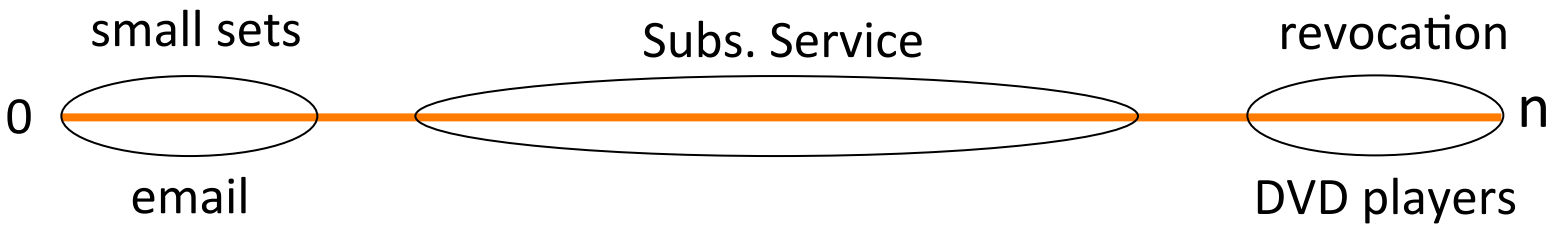


Encrypted mail system:



Social networks: privately send message to a group

Constructions



	|ct|	|sk|	|pk|
The trivial system:	**O(|S|)**	O(1)	O(n)
Revocation schemes: [NNL,HS,GST, LSW,DPP,…]	**O(n-|S|)**	O(log n)	O(1)

Can we have O(1) size ciphertext for all sets S ??

The BGW system: [B-Gentry-Waters'05]	**O(1)**	O(1)	O(n)

The BGW system

Setup(n): $g \leftarrow G$, α, **msk** $\leftarrow Z_q$, def: $g_k = g^{(\alpha^k)}$

$pk = (g, \; g_1, \; g_2, \ldots, g_n, \; g_{n+2}, \ldots, g_{2n}, \; v = g^{msk}) \in G^{2n+1}$

hole

KeyGen(msk, j) $\longrightarrow$ $d_j = (g_j)^{msk} \in G$

Enc(pk, S): $t \leftarrow Z_q$

$ct = \left(g^t, \; (v \cdot \prod_{j \in S} g_{n+1-j})^t \right)$, key $= e(g_n, g_1)^t$

Security

Thm: BGW is statically secure for n users in a bilinear group where n-DDHE assumption holds

n-DDHE: for rand. $g,h \leftarrow G$, $\alpha \leftarrow Z_q$, $R \leftarrow G_2$:

$$[\, h, g, \quad g^{\alpha}, g^{(\alpha^2)}, \ldots, g^{(\alpha^n)}, \quad g^{(\alpha^{n+2})}, \ldots, g^{(\alpha^{2n})}, \quad e(g,h)^{(\alpha^{n+1})} \,]$$

$$\approx_p$$

$$[\, h, g, \quad g^{\alpha}, g^{(\alpha^2)}, \ldots, g^{(\alpha^n)}, \quad g^{(\alpha^{n+2})}, \ldots, g^{(\alpha^{2n})}, \quad R \,]$$

Extensions, Variations, Improvements

Adaptive security: **[GW'10, PPSS'12, …]**

- Adversary can adaptively select what keys to request

Identity-based: **[SF'07, D'07, GW'10, …]**

- Smaller pubic key size: |pk| = O(maximal |S|)

  ⇒ Set of all users can be $\{0, 1, 2, 3, \ldots, 2^{256}\}$

Chosen ciphertext secure: **[BGW'05, PPSS'12, …]**

Trace & revoke: **[BW'06]**

BGW using (log n)-linear map

Recall: BGW Setup(n): $g \leftarrow G$, α, **msk** $\leftarrow Z_q$. pk:

$$g, \quad g^{\alpha}, g^{(\alpha^2)}, \ldots, g^{(\alpha^n)}, \quad g^{(\alpha^{n+2})}, \ldots, g^{(\alpha^{2n})}, \quad v = g^{msk}$$

Suppose: $e_k: G \times \cdots \times G \longrightarrow G_k$; $e: G_k \times G_k \longrightarrow G_{2k}$

Set pk as: (#users $\approx 2^{k-1}$)

$$g, \quad g^{\alpha}, g^{(\alpha^2)}, g^{(\alpha^4)} \ldots, g^{(\alpha^{(2^{2k})})}, g^{(\alpha^{(2^{2k+1})})}, \quad v = g_k^{msk}$$

Using 2k-linear map : can build all needed elements in pk

but for rand. $h \in G$ cannot build $e(g, \ldots, g, h)^{(\alpha^{(2^{2k}-1)})} \in G_{2k}$

BGW using (log n)-linear map

	|ct|	|sk|	|pk|
Bilinear BGW: [B-Gentry-Waters'05]	**O(1)**	O(1)	O(n)
(log n)-linear BGW:	**O(log n)**	O(log n)	$O(\log^2 n)$

<u>Open questions:</u>

- Same parameters without k-linear maps ??
- O(1) size ct from standard lattice assumptions (LWE) ??

Summary

Many open problems in broadcast encryption:

- O(log n) size ciphertext & secret keys from LWE?
- O(log n) size ct, sk, and pub-key w/o k-linear maps?
- Distributed BE with sub-linear ciphertext?