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

PHP TLS binding:

- `fsockopen( "ssl://..." )` : no server-side cert. validation

- **cURL binding**: 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`
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: $e(g^a, g^b) = e(g, g)^{ab} \quad \forall a, b \in \mathbb{Z}, \quad g \in G$
- 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: $sk = \text{rand. } x \in \mathbb{Z}_q$, $pk = g^x \in G$

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

verify($pk$, $m$, $sig$) $\rightarrow$ accept iff $e(g, sig) \overset{?}{=} e(pk, H(m))$

Thm: Existentially unforgeable under CDH in the RO model

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

k-linear map $e: G \times G \times \cdots \times G \rightarrow G_k$ non-degen. & efficient

$e(g, H(m)^x) = e(g^x, H(m))$

Even more applications.

Can they be constructed?
**k-linear maps: a recent breakthrough**

S. Garg, C. Gentry, S. Halevi

**Properties:** (informal)

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

$$
e_1 : G \times G \rightarrow G_2$$
$$
e_2 : G \times G_2 \rightarrow G_3$$
$$
\vdots$$
$$
e_k : G \times G_k \rightarrow G_{k+1}$$

For our purposes:

$$
e_k : G \times \cdots \times G \rightarrow G_k$$
$$
e : G_k \times G_k \rightarrow G_{2k}$$
Broadcast Encryption [Fiat-Naor 1993]

Encrypt to arbitrary subsets $S$:

$c \leftarrow E(pk, S, m)$

$S \subseteq \{1, \ldots, n\}$

d$_1$

d$_2$

d$_3$

Security goal (informal):

Full collusion resistance: secure even if all users in $S^c$ collude
Broadcast Encryption

Public-key BE system:

- \textbf{Setup}(n) \rightarrow \text{pub. key } \text{pk}, \text{ master sec. key } \text{msk}

- \textbf{KeyGen}(\text{msk}, j) \rightarrow d_j \quad (\text{private key for user } j)

- \textbf{Enc}(\text{pk}, S) \rightarrow \text{ct}, k

  \text{k used to encrypt msg for users } S \subseteq \{1, \ldots, n\}

- \textbf{Dec}(\text{pk}, d_j, S, \text{ct}): \text{ If } j \in S, \text{ output } k

Broadcast contains \(( [S], \text{ct}, \text{E}_{\text{SYM}}(k, \text{msg}) ) \)
Broadcast Encryption: Static Security

Semantic security when users collude (static adversary)

Def: \( \text{Adv}[A] = \left| \Pr[\text{b' is correct}] - \frac{1}{2} \right| \)

Security: \( \forall \) poly-time \( A \): \( \text{Adv}[A] \) is negligible
Broadcast systems are everywhere

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

Social networks: privately send message to a group
The trivial system:

- \( |\text{ct}| \): \( O(|S|) \)
- \( |\text{sk}| \): \( O(1) \)
- \( |\text{pk}| \): \( O(n) \)

Revocation schemes:

- \( |\text{ct}| \): \( O(n-|S|) \)
- \( |\text{sk}| \): \( O(\log n) \)
- \( |\text{pk}| \): \( O(1) \)

\[ \text{[NNL, HS, GST, LSW, DPP, ...]} \]

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

The BGW system:

- \( |\text{ct}| \): \( O(1) \)
- \( |\text{sk}| \): \( O(1) \)
- \( |\text{pk}| \): \( O(n) \)

\[ \text{[B-Gentry-Waters’05]} \]
The BGW system

\textbf{Setup}(n): \quad g \leftarrow G, \quad \alpha, \ msk \leftarrow \mathbb{Z}_q, \quad \text{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}

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

\textbf{Enc}(pk, S): \quad t \leftarrow \mathbb{Z}_q

ct = \left( g^t, \ (v \cdot \prod_{j \in S} g_{n+1-j})^t \right) , \quad \text{key} = e(g_n, g_1)^t
Theorem: BGW is statically secure for n users in a bilinear group where n-DDH assumption holds.

n-DDH: for rand. \( g,h \leftarrow G \), \( \alpha \leftarrow Z_q \), \( R \leftarrow G_2 \):

\[
\begin{bmatrix}
  h, g, g^{\alpha}, g^{(\alpha^2)}, \ldots, g^{(\alpha^n)}, g^{(\alpha^{n+2})}, \ldots, g^{(\alpha^{2n})}, e(g,h)^{(\alpha^{n+1})}
\end{bmatrix} \\
\approx_p \\
\begin{bmatrix}
  h, g, g^{\alpha}, g^{(\alpha^2)}, \ldots, g^{(\alpha^n)}, g^{(\alpha^{n+2})}, \ldots, g^{(\alpha^{2n})}, R
\end{bmatrix}
\]
Extensions, Variations, Improvements

Adaptive security: \([GW'10, \text{PPSS'12}, \ldots]\)
• Adversary can adaptively select what keys to request

Identity-based: \([SF'07, D'07, GW'10, \ldots]\)
• Smaller public key size: \(|pk| = O(\text{maximal} \ |S|)\)
  \[\Rightarrow\text{ Set of all users can be } \{0, 1, 2, 3, \ldots, 2^{256}\}\]

Chosen ciphertext secure: \([BGW'05, \text{PPSS'12}, \ldots]\)

Trace & revoke: \([BW'06]\)
BGW using (log n)-linear map

Recall: \textbf{BGW Setup}(n): \ g \leftarrow G, \ \alpha, \ \text{msk} \leftarrow \mathbb{Z}_q . \ \text{pk}:

\[ g, \ g^\alpha, \ g^{(\alpha^2)}, ..., \ g^{(\alpha^n)}, \ g^{(\alpha^{n+2})}, ..., \ g^{(\alpha^{2n})}, \ v=g^\text{msk} \]

Suppose: \( e_k: G \times \cdots \times G \rightarrow G_k \) ; \( e: G_k \times G_k \rightarrow G_{2k} \)

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

\[ g, \ g^\alpha, g^{(\alpha^2)}, g^{(\alpha^4)}, ..., g^{(\alpha^{(2^{2k})})}, g^{(\alpha^{(2^{2k+1})})}, \ v=g^\text{msk} \]

Using 2k-linear map: can build all needed elements in \( \text{pk} \)

but for rand. \( h \in G \) cannot build \( e(g,...,g,h)^{(\alpha^{(2^{2k-1})})} \in G_{2k} \)
**BGW using \((\log n)\)-linear map**

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

Open questions:

- 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?